



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *ACCOUNT PAYABLE* DAN *ACCOUNT RECEIVABLE* YANG BERORIENTASI *MULTI-TENANCY* DENGAN *DISTRIBUTED DATABASE*

SYAFIYUS KURNIAWATI
NRP 5112 100 041

Dosen Pembimbing I
Sarwosri, S.Kom, M.T

Dosen Pembimbing II
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR- K141502

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *ACCOUNT PAYABLE* DAN *ACCOUNT RECEIVABLE* YANG BERORIENTASI *MULTI-TENANCY* DENGAN *DISTRIBUTED DATABASE*

**SYAFIYUS KURNIAWATI
NRP 5112 100 041**

**Dosen Pembimbing I
Sarwosri, S.Kom, M.T.**

**Dosen Pembimbing II
Prof. Drs. Ec. Ir. Rianarto Sarno, M.Sc.,Ph.D.**

JURUSAN TEKNIK INFORMATIKA

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K141502

DESIGN AND IMPLEMENTATION OF ACCOUNT PAYABLE AND ACCOUNT RECEIVABLE SOFTWARE WITH MULTI-TENANCY ORIENTED USING DISTRIBUTED DATABASE

**SYAFIYUS KURNIAWATI
NRP 5112 100 041**

**Supervisor I
Sarwosri, S.Kom, M.T**

**Supervisor II
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *ACCOUNT PAYABLE* DAN *ACCOUNT RECEIVABLE* YANG BERORIENTASI *MULTI-TENANCY* DENGAN *DISTRIBUTED DATABASE*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

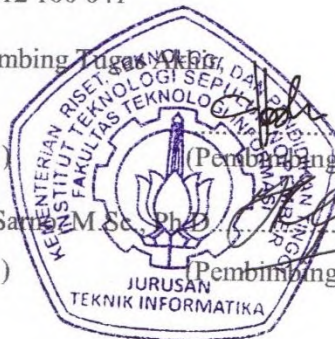
Oleh:

SYAFIYUS KURNIAWATI
NRP 5112 100 041

Disetujui oleh Dosen Pembimbing Tugas Akhir

Sarwosri, S.Kom, M.T.
(NIP. 197608092001122001)

Prof. Drs. Ec. Ir. Riyanarto Saroso, M.Sc., Ph.D.
(NIP. 195908031986011001)



(Pembimbing 1)

(Pembimbing 2)

SURABAYA
JUNI, 2016

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *ACCOUNT PAYABLE* DAN *ACCOUNT RECEIVABLE* YANG BERORIENTASI *MULTI-TENANCY* DENGAN *DISTRIBUTED DATABASE*

Nama Mahasiswa : Syafiyus Kurniawati
NRP : 5112 100 041
Jurusan : Teknik Informatika ITS
Dosen Pembimbing 1 : Sarwosri S.Kom, M.T.
Dosen Pembimbing 2 : Prof. Ir. Drs.Ec. Rryanarto Sarno, M.Sc, Ph.D

ABSTRAK

Enterprise Resource Planning (ERP) adalah sistem informasi perusahaan yang megkoordinasikan semua sumber daya. Dari ERP yang dikembangkan sebelumnya, contoh ERP 2013 menggunakan arsitektur multitenancy separated database yaitu dalam satu sistem ERP, dapat digunakan untuk beberapa perusahaan dengan satu aplikasi yang sama tetapi setiap data perusahaan disimpan di database yang berbeda. Dengan arsitektur tersebut, aplikasi telah menghasilkan fungsionalitas yang optimal, tetapi masih kurang efektif dalam pengelolaan database karena ketika komputer server sedang tidak aktif, maka database tersebut tidak dapat diakses.

Dalam tugas akhir ini, dikembangkan aplikasi ERP yang dibangun dengan menggunakan sistem distributed database berorientasi mutitenancy yaitu satu aplikasi dapat digunakan oleh beberapa perusahaan/tenant secara bersamaan dengan memisahkan database setiap tenant ke dalam komputer/server tersendiri yang dibedakan oleh IP address setiap komputer/server. Hal ini bertujuan untuk mencegah kegagalan sistem jika sebuah komputer/server mengalami down, maka database masih bisa diakses melalui komputer server lain.

Di dalam sistem ERP terdapat bagian yang menangani keuangan khususnya hutang piutang perusahaan yaitu Account Payable dan Account Receivable. Account Payable adalah akun yang mengelola keuangan dalam hal peminjaman hutang dagang kepada pemasok. Account Receivable adalah akun yang mengelola keuangan dalam hal peminjaman piutang dagang kepada pelanggan. Pada modul Account Payable ditambahkan sistem perankingan hutang sesuai dengan tingkat kepentingan hutang. Sedangkan pada Account Receivable ditambahkan sistem perankingan pelanggan berdasarkan tingkat loyalitas pelanggan. Kedua sistem perankingan tersebut menggunakan metode Analytical Hierarchy Process (AHP).

Hasil akhir menunjukkan bahwa sistem ERP yang dirancang menggunakan distributed-database berorientasi multitenancy dapat menyelesaikan masalah ketika salah satu tenant mengalami fail-over dan tenant lain tetap dapat mengakses aplikasi. Penambahan sistem perankingan menggunakan metode AHP pada Account Payable dan Account Receivable dapat memudahkan pengguna dalam pengambilan keputusan.

Kata kunci: Account Payable, Account Receivable, Hutang, Pelanggan, AHP

DESIGN AND IMPLEMENTATION OF ACCOUNT PAYABLE AND ACCOUNT RECEIVABLE MODULE ON ENTERPRISE RESOURCE PLANNING WITH MULTITENANCY ORIENTED USING DISTRIBUTED DATABASE

Student Name : Syafiyus Kurniawati
Student ID : 5112 100 041
Major : Informatics Department FTIf-ITS
Supervisor 1 : Sarwosri, S.Kom, M.T.
Supervisor 2 : Prof. Dr. Eng. Ir. Riyanarto Sarno, M.Sc.,Ph.D.

ABSTRACT

Enterprise Resource Planning (ERP) is a information system to coordinate company resources. In the ERP system developed before, example ERP 2013 used multi-tenancy architecture with separated database which in one system ERP, can used for some enterprise with same application but every enterprise data has saved in different databased. With these architecture, application can run but still not optimal in manage database because when server computer is not active, so database can not accessed.

In this final task, therefor developed an ERP application using multitенancy architecture with distributed database in which one application can used for some enterprise together with separated databased in some server computer. It aim to prevent if computer system has failure or down, database still can accessed through another server computer.

In ERP system, there is a section dealing with financial debt in particular receivable company that is Account Payable and Account Receivable. The Accounts Payable is account to manage finances in terms of loan Payables to suppliers. Accounts Receivable is the accounts that manage finances in terms of the trade accounts receivable lending to customers. In Account

Payable, there are modul that handle on Account Payable module proposed methods to provide weighting or ranked debt in accordance with their importance. On the Accounts Receivable proposed rank customer based on customer purchase history. Both of rank system are used Analytical Hierarchy Process (AHP).

The final results showed that the ERP system designed using a distributed-oriented database multitenancy can solve the problem when one tenant suffered fail-over and other tenant can still access application. The addition of ranking system using the method AHP in Accounts Payable and Accounts Receivable can enabling users in decision making.

Keywords: *ERP (Enterprise Resource Planning), Account Payable, Account Receivable, Debt, Customer, AHP*

KATA PENGANTAR

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga tugas akhir berjudul “Rancang Bangun Sistem *Enterprise Resource Planning* Pada Modul *Account Payable* Dan *Account Receivable* Yang Berorientasi *Multitenancy* Dengan *Distributed Database*” ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu, kakak dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Ibu Sarwosri dan Bapak Riyanarto Sarno selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Segenap dosen rumpun mata kuliah Manajemen Informasi.
6. Teman-teman TC Angkatan 2012, teman TA ERP, dan teman-teman lain yang turut mendukung menyemangati, membantu dan mendengarkan suka duka selama proses pengerjaan tugas akhir.

7. Serta semua pihak yang turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2016

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxvii
DAFTAR KODE SUMBER	xxxi
1. BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	3
1.3. Batasan Permasalahan	4
1.4. Tujuan.....	4
1	5
1.5. Manfaat.....	5
1.6. Metodologi	5
1.7. Sistematika Penulisan.....	6
2. BAB II DASAR TEORI.....	9
2	9
2.1. Penelitian Terkait	9
2.2. <i>ERP (Enterprise Resource Planning)</i>	12
2.2.1. <i>Account Payable</i>	13
2.2.2. <i>Account Receivable</i>	14
2.3. Basis Data Terdistribusi (<i>Distributed Database</i>)	14
2.4. <i>Multitenancy</i>	18
2.5. <i>Proses Bisnis ERP</i>	14
2.5.1. <i>Proses Bisnis Procurement</i>	14
2.5.2. <i>Proses Bisnis Produksi</i>	15
2.5.3. <i>Proses Bisnis Penjualan</i>	15
2.6. Metode AHP (<i>Analytical Hierarchy Process</i>).....	20

2.6.1.	Langkah-langkah penggunaan AHP	20
2.6.2.	Matriks Perbandingan.....	24
2.6.3.	Nilai Total.....	25
2.7.	<i>RBAC (Role Base Access Control)</i>	25
2.8.	<i>PHP (Hypertext Preprocessor)</i>	26
2.9.	Framework Yii2	26
2.10.	MySQL (My Structured Query Language)	26
3.	BAB III ANALISIS DAN PERANCANGAN SISTEM.....	29
3	29	
3.1.	Analisis.....	29
3.1.1.	Analisis Proses Bisnis	29
3.1.2.	Analisis Data	35
3.1.3.	Analisis Metode Perangkingan.....	36
3.2.	Deskripsi Umum Sistem.....	63
3.2.1.	<i>Account Payable</i>	65
3.3.	Identifikasi Pengguna	68
3.4.	Spesifikasi Kebutuhan Perangkat Lunak.....	68
3.4.1.	Kebutuhan Fungsional.....	68
3.5.	Kasus Penggunaan.....	69
3.5.1.	Kasus Penggunaan Mengelola Transaksi Pembelian (UC-001).....	71
3.5.2.	Kasus Penggunaan Mengelola Pembayaran Pembelian (UC-002).....	74
3.5.3.	Kasus Penggunaan Mengelola Transaksi Penjualan (UC-003).....	77
3.5.4.	Kasus Penggunaan Mengelola Pembayaran Penjualan (UC-004).....	80
3.5.5.	Kasus Penggunaan Melakukan Konfirmasi <i>Sales Order</i> (UC-005).....	83

3.5.6.	Kasus Penggunaan Melihat Umur Hutang Perusahaan (UC-006)	85
3.5.7.	Kasus Penggunaan Melihat Umur Piutang Perusahaan (UC-007)	86
3.5.8.	Kasus Penggunaan Melihat Rangking Hutang Perusahaan (UC-008)	88
3.5.9.	Kasus Penggunaan Melihat Rangking Pelanggan (UC-009)	89
3.6.	Perancangan.....	90
3.6.1.	Perancangan <i>Multitenancy</i>	91
3.6.2.	Perancangan Basis Data	92
3.6.3.	Perancangan <i>RBAC (Role Base Access Control)</i>	97
3.6.4.	Perancangan Tampilan	98
4.	BAB IV IMPLEMENTASI SISTEM.....	115
4	115	
4.1.	Lingkungan Pengembangan Sistem	115
4.2.	Implementasi Basis Data Terdistribusi.....	115
4.2.1.	Instalasi Data dan SQL node pada node1 dan node2	116
4.2.2.	Pemasangan Manajemen Node pada node03	117
4.2.3.	Konfigurasi Manajemen Node	118
4.2.4.	Konfigurasi Data dan SQL Node	119
4.2.5.	Memulai MySQL Cluster	119
4.3.	Implementasi RBAC (Role Base Acces Control)	120
4.3.1.	Membuat Tabel Pengguna	120
4.3.2.	Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna	120
4.3.3.	Membuat Modul Admin	121

4.3.4.	Membuat Model Tabel Autentifikasi, <i>Controller</i> dan <i>View</i> Pengguna	121
4.3.5.	Menambahkan Kode pada Kelas <i>usercontroller</i>	121
4.4.	Implementasi Multitenancy	123
4.4.1.	Membuat Halaman Muka Tenant	123
4.4.2.	Menambahkan Database untuk Tenant Baru.....	124
4.4.3.	Login Tenant	124
4.5.	Implementasi Program pada <i>Modul Account Payable</i>	125
4.5.1.	Halaman Utama Modul <i>Account Payable</i>	125
4.5.2.	Melihat Daftar <i>Account Payable Transaction</i> ...	126
4.5.3.	Menambah <i>Account Payable Transaction</i>	127
4.5.4.	Menyunting <i>Account Payable Transaction</i>	128
4.5.5.	Menghapus <i>Account Payable Transaction</i>	129
4.5.6.	Melihat Daftar <i>Account Payable Payment</i>	129
4.5.7.	Menambah <i>Account Payable Payment</i>	130
4.5.8.	Menyunting <i>Account Payable Payment</i>	132
4.5.9.	Menghapus <i>Account Payable Payment</i>	133
4.5.10.	Melihat <i>Account Payable Aging Analysis</i>	134
4.5.11.	Melihat <i>Account Payable Debt Analysis</i>	135
4.6.	Implementasi Program Modul <i>Account Receivable</i> ..	136
4.6.1.	Halaman Utama <i>Account Receivable</i>	136
4.6.2.	Melakukan Konfirmasi <i>Sales Order</i>	137
4.6.3.	Melihat Daftar <i>Account Receivable Transaction</i>	137
4.6.4.	Menambah <i>Account Receivable Transaction</i>	138
4.6.5.	Menyunting <i>Account Receivable Transaction</i> ...	139
4.6.6.	Menghapus <i>Account Receivable Transaction</i>	140

4.6.7.	Melihat Daftar <i>Account Receivable Payment</i>	140
4.6.8.	Menambah <i>Account Receivable Payment</i>	141
4.6.9.	Menyunting <i>Account Receivable Payment</i>	142
4.6.10.	Menghapus <i>Account Receivable Payment</i>	143
4.6.11.	Melihat <i>Account Receivable Aging Analysis</i>	143
4.6.12.	Melihat <i>Account Receivable Customer Analysis</i>	144
4.7.	Implementasi Antarmuka Pengguna <i>Account Payable</i>	145
4.7.1.	Antarmuka Halaman Utama <i>Account Payable</i> ..	145
4.7.2.	Antarmuka Melihat Daftar <i>Account Payable Transaction</i>	145
4.7.3.	Antarmuka Menambah <i>Account Payable Transaction</i>	146
4.7.4.	Antarmuka Menyunting <i>Account Payable Transaction</i>	147
4.7.5.	Antarmuka Menghapus <i>Account Payable Transaction</i>	148
4.7.6.	Antarmuka Melihat Daftar <i>Account Payable Payment</i>	148
4.7.7.	Antarmuka Menambah <i>Account Payable Payment</i>	149
4.7.8.	Antarmuka Menyunting <i>Account Payable Payment</i>	151
4.7.9.	Antarmuka Menghapus <i>Account Payable Payment</i>	151
4.7.10.	Antarmuka Melihat <i>Account Payable Aging Analysis</i>	152
4.7.11.	Antarmuka Melihat <i>Account Payable Debt Analysis</i>	153
4.8.	Implementasi Antarmuka Pengguna <i>Account Receivable</i>	154

4.8.1.	Antarmuka Halaman Utama <i>Account Receivable</i>	154
4.8.2.	Antarmuka Melihat Daftar <i>Account Receivable Transaction</i>	154
4.8.3.	Antarmuka Menambah <i>Account Receivable Transaction</i>	155
4.8.4.	Antarmuka Menyunting <i>Account Receivable Transaction</i>	156
4.8.5.	Antarmuka Menghapus <i>Account Receivable Transaction</i>	157
4.8.6.	Antarmuka Melihat Daftar <i>Account Receivable Payment</i>	157
4.8.7.	Antarmuka Menambah <i>Account Receivable Payment</i>	158
4.8.8.	Antarmuka Menyunting <i>Account Receivable Payment</i>	160
4.8.9.	Antarmuka Menghapus <i>Account Receivable Payment</i>	160
4.8.10.	Antarmuka Melakukan Konfirmasi <i>Sales Order</i>	161
4.8.11.	Antarmuka Melihat Daftar <i>Account Receivable Customer Analysis</i>	161
4.8.12.	Antarmuka Melihat <i>Account Receivable Aging Analysis</i>	162
5.	BAB V PENGUJIAN DAN EVALUASI	163
5		163
5.1.	Lingkungan Pengujian.....	163
5.2.	Skenario Pengujian.....	163
5.2.1.	Pengujian Basis Data Terdistribusi	167
5.2.2.	Pengujian RBAC (Role Based Access Control)	171

5.2.3.	Pengujian <i>Multitenancy</i>	175
5.2.4.	Pengujian Fungsionalitas.....	176
5.3.	Evaluasi Pengujian	201
5.3.1.	Evaluasi Pengujian Fungsionalitas	201
5.3.2.	Evaluasi Pengujian Perhitungan Metode.....	203
BAB VI KESIMPULAN DAN SARAN.....		209
6	209	
6.1.	Kesimpulan.....	209
6.2.	Saran.....	210
LAMPIRAN A – PROSES BISNIS		211
A.1.	Proses Bisnis Odoo.....	211
A.2	Proses Bisnis InoERP	212
A.3	Proses Bisnis Adempiere.....	213
DAFTAR PUSTAKA.....		219
DAFTAR ISTILAH		223
INDEKS		225
BIODATA PENULIS.....		227

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2. 1. Diagram integrasi ERP secara umum	13
Gambar 2. 2. <i>Separated Database</i>	18
Gambar 2. 3. <i>Shared Database, Separated Schema</i>	19
Gambar 2. 4. <i>Shared Database, Shared Schema</i>	19
Gambar 2. 5. Proses Bisnis <i>Procurement</i>	15
Gambar 2. 6. Proses Bisnis Produksi	15
Gambar 2. 7. Proses Bisnis Penjualan	16
Gambar 2. 8. Hierarki Penyelesaian Metode AHP	21
Gambar 3. 1. Proses Bisnis ERP 2016 Level 0	31
Gambar 3. 2. Proses Bisnis <i>Level 1 Account Payable</i>	32
Gambar 3. 3. Proses Bisnis <i>Level 1 Account Receivable</i>	32
Gambar 3. 4. Proses Bisnis <i>Level 2 Account Payable</i>	33
Gambar 3. 5. Proses Bisnis <i>Level 2 Account Receivable</i>	33
Gambar 3. 6. Alur Kerja Sistem Perankingan Hutang (1)	36
Gambar 3. 7. Alur kerja istem Perangkingan Pelanggan (2)	36
Gambar 3. 8. Proses Bisnis <i>Level 3 Account Payable</i>	37
Gambar 3. 9. Proses Bisnis <i>Level 3 Account Receivable</i>	38
Gambar 3. 10. Arsitektur Basis Data Terdistribusi	64
Gambar 3. 11. Alur Kerja <i>Account Payable</i>	65
Gambar 3. 12. Alur kerja <i>Account Receivable</i>	67
Gambar 3. 13. Diagram kasus penggunaan aplikasi	70
Gambar 3. 14. Diagram aktivitas mengelola transaksi pembelian	73
Gambar 3. 15. Diagram aktivitas mengelola pembayaran pembelian	76
Gambar 3. 16. Diagram aktivitas mengelola transaksi penjualan	79
Gambar 3. 17. Diagram aktivitas mengelola pembayaran penjualan	82
Gambar 3. 18. Diagram aktivitas melakukan konfirmasi <i>sales order</i>	84

Gambar 3. 19. Diagram aktivitas melihat umur hutang perusahaan	86
Gambar 3. 20. Diagram aktivitas melihat umur piutang perusahaan	87
Gambar 3. 21. Diagram aktivitas melihat ranking hutang perusahaan	89
Gambar 3. 22. Diagram aktivitas melihat ranking pelanggan	90
Gambar 3. 23. Perancangan <i>Multi-tenancy</i>	91
Gambar 3. 24. Perancangan antarmuka Login	97
Gambar 3. 25. Perancangan antarmuka <i>add user</i>	98
Gambar 3. 26. Rancangan Antarmuka Daftar <i>Account Payable Transaction</i>	99
Gambar 3. 27. Rancangan Antarmuka Membuat <i>Account Payable Transaction</i>	100
Gambar 3. 28. Rancangan Antarmuka Menyunting <i>Account Payable Transaction</i>	101
Gambar 3. 29. Rancangan Antarmuka Menghapus <i>Account Payable Transaction</i>	102
Gambar 3. 30. Rancangan Antarmuka Daftar <i>Account Payable Payment</i>	102
Gambar 3. 31. Rancangan Antarmuka Membuat <i>Account Payable Payment</i>	103
Gambar 3. 32. Rancangan Antarmuka Menyunting <i>Account Payable Payment</i>	104
Gambar 3. 33. Rancangan Antarmuka Menghapus <i>Account Payable Payment</i>	105
Gambar 3. 34. Rancangan Antarmuka Melihat <i>Account Payable Aging Analysis</i>	106
Gambar 3. 35. Rancangan Antarmuka Melihat <i>Account Payable Debt Analysis</i>	106
Gambar 3. 36. Rancangan Antarmuka Melakukan Konfirmasi <i>Sales Order</i>	107
Gambar 3. 37. Rancangan Antarmuka Daftar <i>Account Receivable Transaction</i>	108

Gambar 3. 38. Rancangan Antarmuka Membuat <i>Account Receivable Transaction</i>	108
Gambar 3. 39. Rancangan Antarmuka Menyunting <i>Account Receivable Transaction</i>	109
Gambar 3. 40. Rancangan Antarmuka Menghapus <i>Account Receivable</i>	110
Gambar 3. 41. Rancangan Antarmuka Daftar <i>Account Receivable Payment</i>	111
Gambar 3. 42. Rancangan Antarmuka Membuat <i>Account Receivable Payment</i>	111
Gambar 3. 43. Rancangan Antarmuka Menyunting <i>Account Receivable Payment</i>	112
Gambar 3. 44. Rancangan Antarmuka Menghapus <i>Account Receivable Payment</i>	113
Gambar 3. 45. Rancangan Antarmuka Melihat <i>Account Receivable Aging Analysis</i>	114
Gambar 3. 46. Rancangan Antarmuka Melihat <i>Account Receivable Customer Analysis</i>	114
 Gambar 4. 1. Tampilan Antarmuka Halaman Utama <i>Account Payable</i>	 145
Gambar 4. 2. Tampilan Antarmuka Melihat Daftar <i>Account Payable Transaction</i>	146
Gambar 4. 3. Tampilan Antarmuka Menambah <i>Account Payable Transaction</i>	147
Gambar 4. 4. Tampilan Antarmuka Menyunting <i>Account Payable Transaction</i>	148
Gambar 4. 5. Antarmuka Menghapus <i>Account Payable Transaction</i>	148
Gambar 4. 6. Tampilan Antarmuka Melihat Daftar <i>Account Payable Payment</i>	149
Gambar 4. 7. Tampilan Antarmuka Menambah <i>Account Payable Payment</i>	150
Gambar 4. 8. Antarmuka Menyunting <i>Account Payable Payment</i>	151

Gambar 4. 9. Tampilan Antarmuka Menghapus <i>Account Payable Payment</i>	152
Gambar 4. 10. Tampilan Antarmuka Melihat <i>Account Payable Aging Analysis</i>	152
Gambar 4. 11. Tampilan Antarmuka Melihat <i>Account Payable Debt Analysis</i>	153
Gambar 4. 12. Tampilan Antarmuka Halaman Utama <i>Account Receivable</i>	154
Gambar 4. 13. Tampilan Antarmuka Melihat Daftar <i>Account Receivable Transaction</i>	155
Gambar 4. 14. Tampilan Antarmuka Menambah <i>Account Receivable Transaction</i>	156
Gambar 4. 15. Tampilan Antarmuka Menyunting <i>Account Receivable Transaction</i>	157
Gambar 4. 16. Tampilan Antarmuka Menghapus <i>Account Receivable Transaction</i>	157
Gambar 4. 17. Tampilan Antarmuka Melihat Daftar <i>Account Receivable Payment</i>	158
Gambar 4. 18. Tampilan Antarmuka Menambah <i>Account Receivable Payment</i>	159
Gambar 4. 19. Tampilan Antarmuka Menyunting <i>Account Receivable Payment</i>	160
Gambar 4. 20. Tampilan Antarmuka Menghapus <i>Account Receivable Payment</i>	160
Gambar 4. 21. Antarmuka Melakukan Konfirmasi <i>Sales Order</i>	161
Gambar 4. 22. Tampilan Antarmuka Melihat Daftar <i>Account Receivable Customer Analysis</i>	161
Gambar 4. 23. Tampilan Antarmuka Melihat <i>Account Receivable Aging Analysis</i>	162
 Gambar 5. 1. Pengujian fitur Replikasi pada sistem	168
Gambar 5. 2. Pengujian Fitur Replikasi pada Database Server 1	169
Gambar 5. 3. Pengujian Fitur Replikasi pada Database Server 2	169

Gambar 5. 4. Pengujian Fitur High-Availability pada Sistem...	170
Gambar 5. 5. Pengujian Fitur High-Availability pada Database Server 1	170
Gambar 5. 6. Pengujian Fitur High-Availability pada Database Server 2	170
Gambar 5. 7. Proses login admin	173
Gambar 5. 8. Tampilan awal setelah login berhasil dilakukan..	173
Gambar 5. 9. Proses admin menambahkan user baru.....	174
Gambar 5. 10. Proses menyunting data user oleh admin.....	174
Gambar 5. 11. Proses penghapusan <i>user</i> oleh admin	175
Gambar 5. 12. Data awal Account Payable Transaction	178
Gambar 5. 13. Proses penginputan data <i>Account Payable Transaction</i>	179
Gambar 5. 14. Data <i>Account Payable Transaction</i> yang baru berhasil masuk ke sistem.....	179
Gambar 5. 15. Proses perubahan data <i>Account Payable Transaction</i>	180
Gambar 5. 16. Data <i>Account Payable Transaction</i> berhasil diperbarui	180
Gambar 5. 17. Proses penghapusan <i>Account Payable Transaction</i>	181
Gambar 5. 18. <i>Account Payable Transaction</i> berhasil terhapus	181
Gambar 5. 19. Data awal <i>Account Payable Payment</i>	183
Gambar 5. 20. Proses penginputan data <i>Account Payable Payment</i>	184
Gambar 5. 21. Data <i>Account Payable Payment</i> yang baru berhasil disimpan ke sistem	184
Gambar 5. 22. Perubahan data <i>Account Payable Payment</i>	185
Gambar 5. 23. Data <i>Account Payable Payment</i> berhasil diperbarui	185
Gambar 5. 24. Proses penghapusan <i>Account Payable Payment</i>	186
Gambar 5. 25. <i>Account Payable Payment</i> telah terhapus	186
Gambar 5. 26. Data awal <i>Account Receivable Transaction</i>	188
Gambar 5. 27. Proses penginputan data <i>Account Receivable Transaction</i>	189

Gambar 5. 28. Data <i>Account Receivable Transaction</i> yang baru berhasil disimpan ke sistem.....	189
Gambar 5. 29. Proses perubahan data <i>Account Receivable Transaction</i>	190
Gambar 5. 30. Data <i>Account Receivable Transaction</i> berhasil diperbarui	190
Gambar 5. 31. Proses penghapusan <i>Account Receivable Transaction</i>	191
Gambar 5. 32. <i>Account Receivable Transaction</i> telah terhapus	191
Gambar 5. 33. Proses penginputan data <i>Account Receivable Payment</i>	193
Gambar 5. 34. Data <i>Account Receivable Payment</i> yang berhasil disimpan ke sistem	194
Gambar 5. 35. Proses perubahan data <i>Account Receivable Payment</i>	194
Gambar 5. 36. Data <i>Account Receivable Payment</i> berhasil diperbarui	194
Gambar 5. 37. Proses penghapusan <i>Account Receivable Payment</i>	195
Gambar 5. 38. <i>Account Receivable Payment</i> telah terhapus	195
Gambar 5. 39. <i>Sales Order</i> berhasil disetujui.....	196
Gambar 5. 40. <i>Sales Order</i> berhasil ditolak	197
Gambar 5. 41. Umur hutang perusahaan berhasil ditampilkan .	198
Gambar 5. 42. Umur Piutang Perusahaan berhasil ditampilkan .	199
Gambar 5. 43. Ranking hutang perusahaan berhasil ditampilkan	200
Gambar 5. 44. Ranking pelanggan berhasil ditampilkan	201

DAFTAR TABEL

Tabel 3. 1. Kekurangan dan kelebihan pada Odoo, Adempiere dan InoERP	30
Tabel 3. 2. Deskripsi Proses Bisnis <i>Account Payable Level 2</i>	34
Tabel 3. 3. Deskripsi Proses Bisnis <i>Account Receivable Level 2</i>	34
Tabel 3. 4. Hasil Kuesioner	41
Tabel 3. 5. Pembobotan kuesioner	42
Tabel 3. 6. Pembobotan kriteria	43
Tabel 3. 7. Hasil perhitungan bobot kriteria.....	43
Tabel 3. 8. Nilai prioritas kriteria (Priority vector)	44
Tabel 3. 9. Random Index Saaty	45
Tabel 3. 10. Data transaksi	46
Tabel 3. 11. <i>Pairwise comparison</i> kriteria 1	47
Tabel 3. 12. Normalisasi kriteria 1	47
Tabel 3. 13. <i>Pairwise comparison</i> kriteria 2	47
Tabel 3. 14. Normalisasi kriteria 2	47
Tabel 3. 15. <i>Pairwise comparison</i> kriteria 3	48
Tabel 3. 16. Normalisasi kriteria 3	48
Tabel 3. 17. <i>Pairwise comparison</i> kriteria 4	48
Tabel 3. 18. Normalisasi kriteria 4	48
Tabel 3. 19. <i>Pairwise comparison</i> kriteria 5	49
Tabel 3. 20. Normalisasi kriteria 6	49
Tabel 3. 21. <i>Pairwise comparison</i> kriteria 6	49
Tabel 3. 22. Normalisasi kriteria 6	50
Tabel 3. 23. Tota Skor	50
Tabel 3. 24. Hasil Kuesioner	53
Tabel 3. 25. Pembobotan Kuesioner	54
Tabel 3. 26. Pembobotan kriteria	55
Tabel 3. 27. Hasil Pembobotan Kriteria	56
Tabel 3. 28. Nilai prioritas kriteria (Priority vector)	56
Tabel 3. 29. Data Pelanggan.....	58
Tabel 3. 30. <i>Pairwise comparison</i> kriteria 1	58
Tabel 3. 31. Normalisasi kriteria 1	59

Tabel 3. 32. <i>Pairwise comparison</i> kriteria 2	59
Tabel 3. 33. Normalisasi kriteria 2	59
Tabel 3. 34. <i>Pairwise comparison</i> kriteria 3	60
Tabel 3. 35. Normalisasi kriteria 3	60
Tabel 3. 36. <i>Pairwise comparison</i> kriteria 4	60
Tabel 3. 37. Normalisasi kriteria 4	60
Tabel 3. 38. <i>Pairwise comparison</i> kriteria 5	61
Tabel 3. 39. Normalisasi kriteria 5	61
Tabel 3. 40. <i>Pairwise comparison</i> kriteria 6	61
Tabel 3. 41. Normalisasi kriteria 6	62
Tabel 3. 42. Total Skor Pelanggan	62
Tabel 3. 43. Pengguna Aplikasi.....	68
Tabel 3. 44 Daftar kebutuhan fungsional perangkat lunak.....	69
Tabel 3. 45. Deskripsi Kasus Penggunaan Aplikasi.....	70
Tabel 3. 46. Rincian kasus penggunaan mengelola transaksi pembelian	71
Tabel 3. 47. Rincian kasus penggunaan mengelola pembayaran pembelian	74
Tabel 3. 48. Rincian kasus penggunaan mengelola transaksi penjualan	77
Tabel 3. 49. Rincian kasus penggunaan mengelola pembayaran penjualan	80
Tabel 3. 50. Rincian kasus penggunaan melakukan konfirmasi <i>Sales Order</i>	83
Tabel 3. 51. Rincian kasus penggunaan melihat umur hutang perusahaan.....	85
Tabel 3. 52. Rincian kasus penggunaan melihat umur piutang perusahaan.....	86
Tabel 3. 53. Rincian kasus penggunaan melihat ranking hutang perusahaan.....	88
Tabel 3. 54. Rincian kasus penggunaan melihat ranking pelanggan	89
Tabel 3. 55. Keterangan PDM.....	92

Tabel 5. 1. Pengujian fitur mengelola Role Based Access Control (RBAC)	171
Tabel 5. 2. Pengujian mengelola <i>multitenancy</i>	175
Tabel 5. 3. Pengujian mengelola transaksi pembelian	177
Tabel 5. 4. Pengujian mengelola pembayaran pembelian	181
Tabel 5. 5. Pengujian mengelola transaksi penjualan.....	186
Tabel 5. 6. Pengujian mengelola pembayaran pembelian	191
Tabel 5. 7. Pengujian melakukan konfirmasi <i>sales order</i>	195
Tabel 5. 8. Pengujian melihat umur hutang perusahaan.....	197
Tabel 5. 9. Pengujian melihat umur piutang perusahaan.....	198
Tabel 5. 10. Pengujian melihat ranking hutang perusahaan	199
Tabel 5. 11. Pengujian melihat ranking pelanggan	200
Tabel 5. 12. Rangkuman Hasil Pengujian	202
Tabel 5. 13. Perbandingan hasil perhitungan ranking hutang perusahaan A	203
Tabel 5. 14. Perbandingan hasil perhitungan ranking hutang perusahaan B	204
Tabel 5. 15. Perbandingan hasil perhitungan ranking hutang perusahaan C	204
Tabel 5. 16. Perbandingan hasil perhitungan ranking pelanggan Perusahaan A.....	206
Tabel 5. 17. Perbandingan hasil perhitungan ranking pelanggan Perusahaan B	206
Tabel 5. 18. Perbandingan hasil perhitungan ranking pelanggan Perusahaan C	207

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1. Membuat grup MySQL pengguna baru dan menambah user MySQL.....	116
Kode Sumber 4.2. Mengubah Lokasi Direktori, Mengubah Arsip, dan Menciptakan <i>Symlink</i>	116
Kode Sumber 4.3. Mengubah Lokasi Direktori	116
Kode Sumber 4.4. Mengatur Perizinan <i>Server</i> MySQL.....	117
Kode Sumber 4.5. Menyalin script startup MySQL.....	117
Kode Sumber 4. 6. Mengubah Lokasi Direktori	117
Kode Sumber 4. 7. Mengubah Lokasi Direktori	118
Kode Sumber 4. 8. Membuat Direktori.....	118
Kode Sumber 4. 9. Mengatur file "config.ini"	118
Kode Sumber 4. 10. Data dan SQL Node	119
Kode Sumber 4. 11. Memulai Proses Manajemen Node.....	119
Kode Sumber 4. 12. Memulai Proses ndbd dan MySQL Server	119
Kode Sumber 4. 13. Mengaktifkan MySQL	120
Kode Sumber 4.14. Generate Tabel Autentifikasi.....	120
Kode Sumber 4. 15. Pembuatan Modul Admin	121
Kode Sumber 4. 16. Kode Fungsi Tabel AuthItem	122
Kode Sumber 4. 17. Kode Fungsi Tabel ItemChild	122
Kode Sumber 4. 18. Kode Fungsi Tabel AuthAssignment	123
Kode Sumber 4. 19. Pembuatan Halaman Muka Tenant	124
Kode Sumber 4. 20. Penambahan Database Tenant Baru	124
Kode Sumber 4. 21. Login Tenant	125
Kode Sumber 4. 22. Halaman Utama Modul <i>Account Payable</i>	126
Kode Sumber 4.23. Melihat Daftar <i>Account Payable Transaction</i>	127
Kode Sumber 4.24. Menambah <i>Account Payable Transaction</i>	128
Kode Sumber 4. 25. Menyunting <i>Account Payable Transaction</i>	129
Kode Sumber 4. 26. Menghapus <i>Account Payable Transaction</i>	129

Kode Sumber 4. 27. Melihat Daftar <i>Account Payable Payment</i>	130
Kode Sumber 4. 28. Menambah <i>Account Payable Payment</i>	131
Kode Sumber 4. 29. Menyunting <i>Account Payable Payment</i> ...	133
Kode Sumber 4. 30. Menghapus <i>Account Payable Payment</i>	134
Kode Sumber 4. 31. Melihat <i>Account Payable Aging Analysis</i>	134
Kode Sumber 4. 32. Melihat <i>Account Payable Debt Analysis</i> ..	135
Kode Sumber 4. 33. Halaman Utama <i>Account Receivable</i>	136
Kode Sumber 4. 34. Melakukan Konfirmasi <i>Sales Order</i>	137
Kode Sumber 4. 35. Melihat Daftar <i>Account Receivable Transaction</i>	138
Kode Sumber 4. 36. Menambah <i>Account Receivable Transaction</i>	138
Kode Sumber 4. 37. Menyunting <i>Account Receivable Transaction</i>	139
Kode Sumber 4. 38. Menghapus <i>Account Receivable Transaction</i>	140
Kode Sumber 4. 39. Melihat Daftar <i>Account Receivable Payment</i>	140
Kode Sumber 4. 40. Menambah <i>Account Receivable Payment</i>	141
Kode Sumber 4. 41. Menyunting <i>Account Receivable Payment</i>	142
Kode Sumber 4. 42. Menghapus <i>Account Receivable Payment</i>	143
Kode Sumber 4. 43. Melihat <i>Account Receivable Aging Analysis</i>	143
Kode Sumber 4. 44. Melihat <i>Account Receivable Customer Analysis</i>	144

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pengerjaan Tugas Akhir, dan sistematika penulisan Tugas Akhir.

1.1. Latar Belakang

Enterprise Resource Planning (ERP) adalah sistem informasi perusahaan yang megkoordinasikan semua sumber daya. Di dalam sistem ERP terdapat bagian yang menangani keuangan khususnya hutang piutang perusahaan yaitu *Account Payable* dan *Account Receivable*.

Account Payable merupakan hutang usaha sebuah perusahaan, dapat diartikan sebagai kewajiban perusahaan untuk membayar pembelian barang atau jasa yang telah dilakukan kepada pemasok atau *supplier*. Pencatatan hutang dagang ini sangat penting untuk melacak hutang perusahaan di waktu tertentu. Dengan demikian perusahaan dapat mengetahui berapa banyak hutang yang harus dibayar untuk setiap barang yang masuk dan waktu jatuh tempo pembayaran. Permasalahan yang muncul pada *Account Payable* adalah adanya keterbatasan *cash flow* perusahaan dalam menjalankan kegiatan operasional yang berpengaruh terhadap produktivitas perusahaan. Sehingga perusahaan di tuntut untuk dapat mengembangkan strategi dan kebijakan yang tepat agar permasalahan mengenai *Account Payable* tersebut dapat diatasi.

Account Receivable adalah piutang usaha yang merupakan hasil penjualan barang atau jasa perusahaan yang didapat dari pelanggan atau *customer*. *Account Receivable* ini muncul karena adanya penjualan secara kredit. Penjualan secara kredit bertujuan untuk meningkatkan daya tarik agar dapat mempertahankan pelanggan, sehingga penjualan dan laba perusahaan dapat

ditingkatkan. Permasalahan yang muncul pada *Account Receivable* adalah adanya keterbatasan *cash flow* akibat kelonggaran dalam menarik piutang perusahaan kepada pelanggan. Perusahaan mengharapkan dapat menerima piutang dagangnya sesuai dengan jatuh tempo kredit. Sehingga perusahaan diuntut untuk dapat mengembangkan strategi dan kebijakan yang tepat agar permasalahan mengenai *Account Receivable* tersebut dapat diatasi.

Pada pengembangan aplikasi ERP 2013, modul *Account Payable* dan *Account Receivable* terdapat sub-modul *Purchase Invoice* dan *Purchase Return Invoice*. Sub-modul tersebut belum cukup lengkap jika dibandingkan dengan aplikasi ERP yang banyak digunakan seperti SAP. Aplikasi dikembangkan dengan arsitektur berorientasi layanan atau SOA (*Service Oriented Architecture*) menggunakan sistem MVC (*Model-View-Controller*) dan *Workflow* untuk .NET. Sistem database yang digunakan adalah *separated database* yaitu dalam satu sistem ERP, setiap data perusahaan disimpan di *database* yang berbeda. Sistem *database* tersebut belum cukup efektif karena akan memerlukan banyak *database*.

Pada tugas akhir ini, dikembangkan aplikasi ERP pada modul *Account Payable* dan *Account Receivable* yang memiliki sub-modul *AP Transaction*, *AP Payment*, *AP Report*, *AR Transaction*, *AR Payment*, *AR Report*. Kemudian ditambahkan sub-modul baru yaitu *AP Aging*, *AP Debt Analysis*, *AR Aging* dan *AR Customer Analysis*.

Sub-modul *AP Aging* merupakan laporan jumlah hutang perusahaan pada periode tertentu. *AP Debt Analysis* berisi rekomendasi urutan hutang sesuai dengan prioritas kepentingan.

Perangkingan hutang menggunakan salah satu metode sistem pendukung keputusan yaitu metode AHP (*Analytical Hierarchy Process*). Data pada *AP Debt Analysis* dapat digunakan oleh *Account Payable staff* untuk melihat rekomendasi hutang yang sebaiknya dibayar terlebih dahulu dan hutang yang dapat ditunda pembayarannya.

Sub-modul *AR Aging* merupakan report piutang perusahaan pada periode tertentu. *AR Customer Analysis* berisi penilaian terhadap *customer* yang telah melakukan transaksi pembelian dengan perusahaan. Perangkingan hutang menggunakan salah satu metode sistem pendukung keputusan yaitu metode AHP (*Analytical Hierarchy Process*). Data pada *AR Customer Analysis* dapat digunakan oleh bagian penjualan dan bagian keuangan untuk mengambil kebijakan terhadap pelanggan yang memiliki catatan pembelian yang buruk.

Aplikasi dibangun menggunakan arsitektur MVC (*Model-View-Controller*) dengan *database* yang bersistem *Multi-Tenancy Separated Database* pada sistem basis data terdistribusi (*distributed database*). Mekanisme pada *Multi-Tenancy Separated Database*, secara fisik data setiap *tenant* disimpan terpisah dengan *tenant* lainnya.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana memodelkan *business process* yang meliputi sub-modul *AP Transaction*, *AP Payment*, *AR Transaction*, *AP Report*, *AP Debt Analysis*, *AR Transaction*, *AR Payment*, *AR Report*, *AR Customer Analysis* terkait modul *Account Payable* dan *Account Receivable* yang terdapat pada sistem ERP?
2. Bagaimana mengimplementasikan *multi-tenancy separated database* pada *distributed database* pada sistem ERP?
3. Bagaimana mengimplementasikan sistem perangkingan hutang dan *customer* dengan menggunakan metode AHP?

1.3. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, antara lain:

1. Hasil dari tugas akhir ini adalah sebuah aplikasi ERP modul *Account Payable* dan *Account Receivable*.
2. Sub-modul *Account Payable* antara lain: *AP Transaction*, *AP Payment*, *AP Aging*, *AP Debt Analysis*.
3. Sub-modul pada modul *Account Receivable* antara lain: *AR Transaction*, *AR Payment*, *AR Aging*, *AR Customer Analysis*..
4. Sistem perankingan hutang dan pelanggan menggunakan metode AHP (*Analytical Hierarchy Process*) yang bersifat statis.
5. Sistem ERP yang dibangun dalam aspek *Multitenancy* dengan *distributed database*.
6. Platform menggunakan framework yii2 dengan teknologi RBAC (*Role Base Access Control*)

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Memodelkan *business process* model yang meliputi sub-modul *AP Transaction*, *AP Payment*, *AR Transaction*, *AP Report*, *AP Debt Analysis*, *AR Transaction*, *AR Payment*, *AR Report*, *AR Customer Analysis* terkait modul *Account Payable* dan *Account Receivable* yang terdapat pada sistem ERP.
2. Mengimplementasikan *multi-tenancy separated database* pada *distributed database* pada sistem ERP.
3. Mengimplementasikan sistem perankingan hutang dan customer dengan menggunakan metode AHP.

1.5. Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah memudahkan *Account Payable* dan *Account Receivable* perusahaan dalam pengelolaan keuangan perusahaan. Dengan sistem keuangan perusahaan yang dikelola dengan baik dan terintegrasi, diharapkan kinerja dan pendapatan perusahaan dapat meningkat.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Pada tahapan ini akan dilakukan studi literatur yang berfokus pada pembuatan aplikasi PHP dengan *framework* Yii 2.0 sesuai dengan proses bisnis model dari ERP, implementasi *multitenancy* dengan *distributed database* serta proses bisnis yang akan digunakan seperti *Account Payable*, *Account Receivable*.

2. Analisis dan desain perangkat lunak

Pada tahap analisis dan desain dilakukan perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang digunakan. Tahap ini mendefinisikan alur dari implementasi, perancangan database, perancangan *prototype system*. Desain perangkat lunak akan menggunakan sistem rancang MVC (*Model-View-Controller*).

3. Implementasi perangkat lunak

Tahap implementasi dilakukan dengan cara menerapkan algoritma-algoritma pada sistem dibangun menggunakan bahasa pemrograman PHP dengan Framework Yii 2.0. Pengolahan database menggunakan database MySQL.

4. Pengujian dan evaluasi

Tahap pengujian dilakukan dengan menjalankan program menggunakan 3 komputer yang menjalankan sistem sebagai tenant dan menguji kemampuan *multitenant* dengan mematikan salah satu *computer server* dan memastikan apakah tenant yang lain masih mampu menjalankan proses bisnisnya tanpa terpengaruh dari tenant lain.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1. Penelitian Terkait

Tugas Akhir ini merupakan pengembangan berkelanjutan dari riset atau penelitian tentang ERP. Pada Tugas Akhir yang sebelumnya, telah dibangun sebuah aplikasi bernama ERP 2013. Pada aplikasi ERP 2013, aplikasi dikembangkan dengan sistem *multitenancy* yaitu dalam satu sistem ERP, dapat digunakan untuk beberapa perusahaan dengan satu aplikasi yang sama sehingga lebih efektif dalam penggunaan memori komputer server. Sistem database yang digunakan adalah *separated database*, yaitu setiap data perusahaan disimpan di *database* yang berbeda. Dengan sistem *separated database* tersebut, aplikasi telah menghasilkan fungsionalitas yang optimal, tetapi masih kurang efektif dalam pengelolaan *database*. Hal itu dikarenakan ketika komputer server sedang tidak aktif atau mati, maka *database* tersebut tidak dapat diakses.

Pada aplikasi ERP 2013 terdapat Modul *General Ledger/Cash Bank, Account Payable/ Account Receivable, Supplier Relationship Management, Inventory, Production Execution, Customer Relationship Management*, dan *Production Planning*. Masing-masing modul memiliki alur keterkaitan dengan modul yang lain. Data pada setiap tenant/perusahaan disimpan pada *database* yang berbeda atau *separated database*.

Pada tugas akhir ini, dikembangkan aplikasi ERP yang bernama ERP 2016, sistem dibangun dengan menggunakan sistem *distributed database* yaitu memisahkan *database* setiap *tenant* ke dalam komputer/*server* tersendiri yang dibedakan oleh *IP address* setiap komputer/*server*. Hal ini bertujuan

untuk mencegah kegagalan sistem jika sebuah komputer/server mengalami *down*. Maka *database* masih bisa diakses melalui komputer server lain.

Modul pada ERP 2016 terdiri dari Modul *General Ledger*, *Account Payable* and *Account Receivable*, *Inventory and Warehouse Management*, *Production*, *Sales and Distribution*, *Finance*, *Purchasing*, *Asset Management* dan *Human Resource*. Masing-masing modul memiliki alur keterkaitan dengan modul yang lain. Perbedaan modul yang terdapat pada ERP 2013 dan ERP 2016 ditunjukkan pada Tabel 2.1.

Tabel 2. 1. Perbedaan Modul ERP 2013 dan 2016

No.	Nama Modul ERP 2013	Nama Modul ERP 2016
1	<i>Account Payable & Receivable</i>	<i>Account Payable & Receivable</i>
2	<i>General Ledger</i>	<i>General Ledger</i>
3	<i>Cost Accounting</i>	<i>Finance</i>
4	<i>Customer Relationship Management</i>	<i>Sales and Distribution</i>
5	<i>Production Planning</i>	<i>Production</i>
6	<i>Supplier Relationship Management</i>	<i>Purchasing</i>
7	<i>Inventory</i>	<i>Inventory and Warehouse Management</i>
8	<i>Assets Management</i>	<i>Assets Management</i>
9	-	<i>Human Resource Management</i>

Kemudian pada bagian keuangan, khususnya modul *Account Payable* dan *Account Receivable* aplikasi ERP 2013 terdapat sub-modul *Purchase Invoice*, *Purchase Return Invoice*, *Sales Invoice* dan *Sales Return Invoice*. Sub-modul tersebut belum cukup lengkap jika dibandingkan dengan aplikasi ERP yang banyak digunakan seperti SAP. Maka pada aplikasi ERP 2015, dikembangkan modul *Account Payable* dan *Account Receivable* yang memiliki sub-modul *AP Transaction*, *AP*

Payment, AR Transaction, AR Payment. AP Transaction dan AP Payment memiliki fungsi yang sama dengan modul *Purchase Invoice*, sedangkan *AR Transaction dan AR Payment* memiliki fungsi yang sama dengan *Sales Invoice dan Sales Return Invoice* pada ERP 2013. Kemudian ditambahkan sub-modul baru yaitu *AP Aging, AP Debt Analysis, AR Aging dan AR Customer Analysis*. Berikut perbedaan antara ERP 2013 dan ERP 2016 pada modul *Account Payable dan Account Receivable* ditunjukkan pada Tabel 2.2.

Tabel 2. 2. Perbedaan Modul Account Payable dan Account Receivable ERP 2013 dan ERP 2016

No	ERP 2013	ERP 2016
1	<i>Purchase Invoice</i>	<i>AP Transaction</i>
2	<i>Purchase Return Invoice</i>	<i>AP Payment</i>
3	<i>Sales Invoice</i>	<i>AR Transaction</i>
4	<i>Sales Return Invoice</i>	<i>AR Payment</i>
5	-	<i>AP Aging</i>
6	-	<i>AP Debt Analysis</i>
7	-	<i>AR Aging</i>
8	-	<i>AR Customer Analysis</i>

Sub-modul *AP Aging* merupakan laporan jumlah hutang perusahaan pada periode tertentu. *AP Debt Analysis* berisi rekomendasi urutan hutang sesuai dengan prioritas kepentingan. Perangkingan hutang menggunakan salah satu metode sistem pendukung keputusan yaitu metode AHP (*Analytical Hierarchy Process*). Data pada *AP Debt Analysis* dapat digunakan oleh *Account Payable staff* untuk melihat rekomendasi hutang yang sebaiknya dibayar terlebih dahulu dan hutang yang dapat ditunda pembayarannya.

Sub-modul *AR Aging* merupakan laporan piutang perusahaan pada periode tertentu. *AR Customer Analysis* berisi penilaian terhadap pelanggan yang telah melakukan transaksi

pembelian dengan perusahaan. Perangkingan pelanggan menggunakan salah satu metode sistem pendukung keputusan yaitu metode AHP (*Analytical Hierarchy Process*). Data pada *AR Customer Analysis* dapat digunakan oleh bagian penjualan (*sales*) dan bagian keuangan (*finance*) untuk mengambil kebijakan terhadap pelanggan yang memiliki catatan pembelian yang buruk.

2.2. ERP (*Enterprise Resource Planning*)

ERP / Perencanaan Sumber Daya Perusahaan adalah sistem informasi yang dirancang dan digunakan perusahaan yang bergerak di bidang manufakturing maupun bidang jasa yang berperan mengintegrasikan dan mengotomasikan proses bisnis yang berhubungan dengan aspek produksi, operasi, manusia, mesin, dana, material, suku cadang, waktu, bahan baku dan kapasitas maupun distribusi perusahaan tersebut [1].

Keuntungan penggunaan ERP dalam perusahaan diantaranya adalah adanya integrasi standarisasi data dan informasi, penurunan inventori dan tenaga kerja, data keuangan, standarisasi proses operasi, peningkatan layanan dan kontrol keuangan dan penurunan waktu yang dibutuhkan untuk mendapatkan informasi [2].

Ada pun departemen-departemen atau bagian-bagian yang pada umumnya terintegrasi meliputi:

1. Pengelolaan keuangan (*financial management*) dalam aplikasi ERP 2016 ini diwakili oleh modul *Account Payable*, *Account Receivable*, *Fixed Asset*, *Cost Management*, *General Ledger*
2. Pengelolaan rantai pemasokan (*supply chain management*) diwakili oleh modul *Purchasing*
3. Perencanaan produksi dan manufaktur (*manufacturing resource planning*) diwakili oleh modul *Production* dan *Inventory*
4. Pengelolaan sumber daya manusia (*human resource management*) diwakili oleh modul *Human Resource*.

5. Pengelolaan relasi dengan pelanggan (*customer relationship management*) diwakili oleh modul *Sales Order*.

Departemen-departemen tersebut diintegrasikan melalui suatu sistem ERP terpusat, seperti yang ditunjukkan dalam Gambar 2.1



Gambar 2. 1. Diagram integrasi ERP secara umum

Pada tugas akhir ini, akan dibangun dua modul dari domain/modul fungsi yang termasuk dalam sistem ERP, yaitu *Account Payable* (Hutang Dagang) dan *Account Receivable* (Piutang Dagang). Penjelasan lebih mendetail lagi mengenai masing-masing domain adalah sebagai berikut.

2.2.1. *Account Payable*

Account Payable (Hutang Dagang) adalah sebuah akun yang mencatat jumlah hutang yang harus dibayarkan oleh perusahaan kepada pemasok atau *supplier*. Apabila perusahaan

telah menerima tagihan pembayaran dari *supplier*, maka tagihan tersebut akan ditambahkan ke dalam akun hutang dagang atau *account payable*, dan kemudian akan dicatat pembayarannya ketika hutang tersebut telah dibayarkan.

2.2.2. Account Receivable

Account Receivable (Piutang Dagang) adalah jumlah uang yang harus dibayarkan oleh pelanggan kepada perusahaan karena membeli barang atau jasa secara kredit yang nantinya akan menjadi pemasukan bagi keuangan perusahaan. Proses pencatatan piutang dagang biasanya dilakukan dengan membuat faktur yang nantinya akan dikirimkan kepada pelanggan, dan kemudian harus dibayarkan dalam jangka waktu yang telah ditentukan.

2.3. Proses Bisnis ERP

Sebelum menggunakan aplikasi ERP, maka pengguna harus memahami proses bisnis dalam satu perusahaan. Hal ini harus dikuasai karena aliran aplikasi ERP mengikuti proses bisnis suatu perusahaan. Ada beberapa proses bisnis perusahaan yang harus dianalisis yaitu pengadaan (*procurement*), produksi (*production*), penjualan (*sales/fullfillment*).

2.3.1. Proses Bisnis Procurement

Proses bisnis pengadaan merupakan proses dimana suatu perusahaan membeli barang untuk diproduksi. Proses bisnis ini dimulai dengan perusahaan membuat dokumen *purchase requisition* (PR). Kemudian dokumen PR ini setelah disetujui akan menjadi dokumen *purchase order* (PO) atau perintah pembelian barang. Setelah barang sudah dibeli, maka muncul dokumen *receive material* (RM) yang menandakan barang sudah diterima. Kemudian perusahaan mendapatkan tagihan atau *invoice* untuk dibayar. Terakhir perusahaan membayarkan sejumlah uang untuk melunasi tagihan tersebut. Terdapat Gambar 2.5 untuk menjelaskan dan menggambarkan proses bisnis *procurement*.



Gambar 2. 2. Proses Bisnis *Procurement*

2.3.2. Proses Bisnis Produksi

Proses bisnis produksi merupakan proses perusahaan memproduksi barang. Proses business ini dimulai dengan perusahaan membuat dokumen *request production*. Setelah produksi disetujui maka akan muncul dokumen *authorized production*. Produksi akan dapat berjalan ketika bahan mentah sudah tersedia dan dapat dikeluarkan dari gudang. Proses produksi dimulai ketika dokumen *create product* sudah muncul. Setelah barang sudah jadi maka, barang akan dipindahkan dari tempat produksi ke gudang dengan adanya dokumen *receive finished goods*. Terdapat Gambar 2.6. untuk menjelaskan dan menggambarkan proses bisnis produksi.



Gambar 2. 3. Proses Bisnis Produksi

2.3.3. Proses Bisnis Penjualan

Proses bisnis penjualan merupakan proses perusahaan menjual barang kepada pelanggan.. Proses bisnis ini dimulai dengan perusahaan membuat dokumen *sales order* untuk mengelola siapa yang membeli barang, barang yang dibeli, jumlah barang, harga barang, dan total harga. Kemudian proses dilanjutkan dengan mempersiapkan pengiriman barang. Barang

akan dikirim dan muncul dokumen *shipment*. Setelah barang diterima oleh *customer*, maka perusahaan mengirimkan tagihan atau *invoice* kepada pelanggan. Proses bisnis ini berakhir ketika pelanggan sudah membayar tagihan tersebut. Proses bisnis produksi ditunjukkan pada Gambar 2.7.



Gambar 2. 4. Proses Bisnis Penjualan

2.4. Basis Data Terdistribusi (*Distributed Database*)

Basis data terdistribusi (*distributed database*) adalah suatu basis data yang berada di bawah kendali sistem manajemen basis data (DBMS) terpusat dengan piranti penyimpanan (*storage devices*) yang terpisah-pisah satu dari yang lainnya. Tempat penyimpanan ini dapat berada di satu lokasi yang secara fisik berdekatan atau terpisah oleh jarak yang jauh dan terhubung melalui jaringan internet. Penggunaan basis data terdistribusi dapat dilakukan di server internet, intranet atau ekstranet kantor, atau di jaringan perusahaan.

Terdapat beberapa konsep dalam basis data terdistribusi, antara lain:

1. *Database Cluster*

Database clustering adalah kumpulan dari beberapa server yang berdiri sendiri yang kemudian bekerjasama sebagai suatu sistem tunggal [3]. Aplikasi database semakin berkembang, baik dalam hal kegunaan, ukuran, maupun kompleksitas berdampak pada server *database* sebagai penyedia layanan terhadap akses *database*. Beban *database*

server akan semakin bertambah berat dan mengakibatkan kinerja *server* tersebut kurang optimal. Oleh karena itu diperlukan perancangan yang tepat dan handal dan tepat dalam membangun *database server*.

Saat aplikasi dijalankan, *Clustering database* yang disimpan dapat terbagi ke beberapa perangkat, semua perangkat yang menyimpan data tersebut dianggap sebagai satu kesatuan. Metode clustering seperti ini sangat baik untuk *loadbalancing* dan penanganan *system failure* karena kemampuan tiap perangkat akan digunakan dan jika ada salah satu perangkat yang mengalami *failure* maka sistem tidak akan langsung terganggu karena perangkat lain akan tetap berfungsi. Kemampuan *clustering* memungkinkan sebuah *database* tetap hidup dalam waktu yang lama.

2. ***MySQL Cluster***

MySQL Cluster merupakan sebuah tipe basis data yang dapat beroperasi dalam ukuran data yang relatif besar (mencapai 100 gigabyte). *MySQL Cluster* adalah sebuah teknologi baru untuk memungkinkan clustering di dalam memory database dalam sebuah sistem *share-nothing* [3]. Arsitektur *share-nothing* memungkinkan sistem dapat bekerja dengan hardware/perangkat keras yang sangat murah, dan tidak membutuhkan perangkat keras dan lunak dengan spesifikasi khusus. Arsitektur tersebut juga handal karena masing-masing komponen mempunyai memory dan disk tersendiri. *MySQL Cluster* menggabungkan *MySQL Server* biasa dengan sebuah mesin penyimpanan in-memory tercluster yang dinamakan NDB. NDB berarti bagian dari suatu rangkaian yang dikhususkan sebagai mesin penyimpanan, sedangkan *MySQL Cluster* diartikan sebagai kombinasi atau gabungan dari *MySQL* dan mesin penyimpanan yang baru tersebut.

3. ***Arsitektur MySQL Cluster***

Arsitektur *MySQL Cluster* adalah arsitektur *shared-nothing* dengan antarmuka SQL yang telah umum digunakan [3].

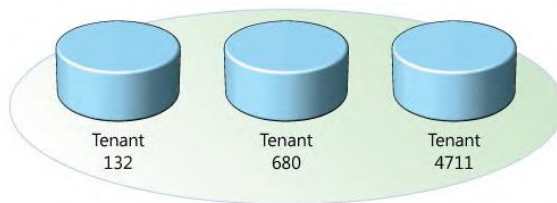
Sistem database ini terdiri dari beberapa node yang dapat didistribusikan ke beberapa perangkat keras dan ke beberapa wilayah/zona yang berbeda sekaligus untuk tetap menjaga ketersediaan data meskipun jaringan ataupun salah satu node sedang mengalami kegagalan (*failure*).

2.5. *Multitenancy*

Multitenancy merupakan suatu prinsip dari arsitektur perangkat lunak, dimana sebuah perangkat lunak yang berjalan di atas server melayani banyak pengguna/tenant. Dengan prinsip multi tenant ini, sebuah perangkat lunak dirancang untuk memiliki partisi data yang berbeda dan dapat dikonfigurasi [4].

Arsitektur data *Multitenancy* digunakan karena dinilai memadai dan cukup aman dalam mengatasi masalah kurang kepercayaan *tenant* untuk menyerahkan kontrol data bisnis *tenant* kepada pihak ketiga. Terdapat 3 jenis *Multitenancy*, antara lain:

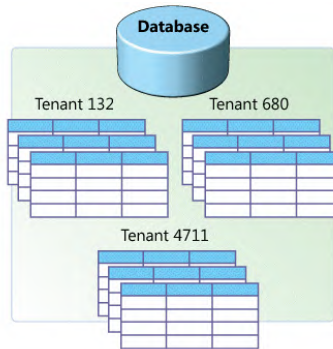
1. ***Separated Database*** yaitu data setiap *tenant* disimpan pada *database* yang terpisah dengan *tenant* lain. Keuntungan arsitektur ini adalah mudah untuk mengatur kembali model data aplikasi yang digunakan. Tetapi memerlukan biaya yang cukup tinggi untuk menjaga peralatan *server* dan juga *back up* data dari setiap *tenant*. Model arsitektur *Separated Database* ditunjukkan pada Gambar 2.2.



Gambar 2. 5. *Separated Database*

2. ***Shared Database, Separated Schema*** yaitu data setiap *tenant* disimpan pada satu *database* tetapi pada *schema*

yang terpisah dengan *tenant* lain. Keuntungan dari arsitektur ini adalah mudah untuk digunakan karena tabel yang dibentuk pada awalnya merupakan tabel standar, dan selanjutnya dapat diubah sesuai keinginan *tenant*. Akan tetapi apabila terjadi kegagalan maka perlu dilakukan perbaikan untuk semua tabel yang ada dalam *database*. Model arsitektur *Shared Database, Separated Schema* ditunjukkan pada Gambar 2.3.



Gambar 2. 6. *Shared Database, Separated Schema*

3. ***Shared Database, Shared Schema*** yaitu data setiap *tenant* disimpan pada satu *database* dan satu *schema*. Keuntungan dari arsitektur ini adalah tidak memerlukan biaya yang tinggi, akan tetapi apabila terjadi kegagalan maka perlu dilakukan perbaikan untuk semua tabel yang ada dalam *database*. Model arsitektur *Shared Database, Shared Schema* ditunjukkan pada Gambar 2.4.

	TenantID	CustName	Address
4	TenantID	ProductID	ProductName
1	4	TenantID	Shipment
6	1	4711	324965
4	6	132	115468
	4	680	654109
		4711	324956
			2006-02-21
			2006-04-08
			2006-03-27
			2006-02-23

Gambar 2. 7. *Shared Database, Shared Schema*

2.6. Metode AHP (*Analytical Hierarchy Process*)

AHP merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty [5]. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki. AHP sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut [6] :

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan.
3. Memperhitungkan daya tahan output analisis sensitivitas pengambilan keputusan

Selain itu metode AHP mempunyai kemampuan untuk memecahkan masalah yang multi-objektif dan multikriteria yang berdasar pada perbandingan preferensi dari setiap elemen dalam hirarki. Jadi metode AHP merupakan suatu bentuk pemodelan pembuatan keputusan yang sangat komprehensif.

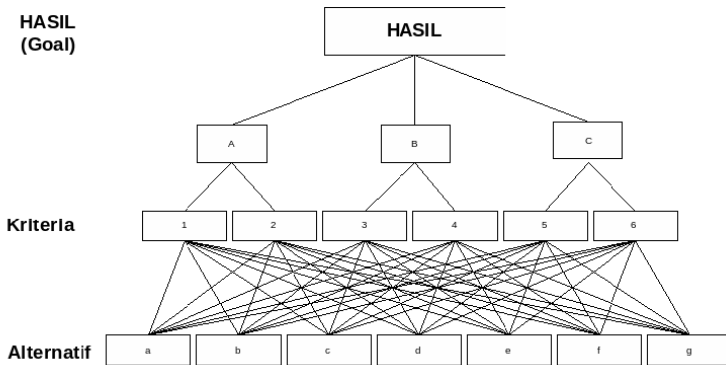
2.6.1. Langkah-langkah penggunaan AHP

Pada dasarnya terdapat beberapa langkah yang perlu diperhatikan dalam menggunakan metode AHP sebagai berikut :

2.6.1.1. Pemodelan

Pada tahap pemodelan, setelah persoalan didefinisikan, perlu dilakukan dekomposisi, yaitu: memecahkan masalah persoalan yang utuh menjadi unsur – unsurnya. Jika ingin mendapatkan hasil yang akurat, pemecahan juga dilakukan terhadap unsur - unsurnya sampai tidak mungkin lagi dilakukan pemecahan lebih lanjut, sehingga didapat beberapa tingkatan dari persoalan tadi, dimana level pertama adalah tujuan (hasil), yang diikuti level kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir adalah alternatif. Karena alasan ini, maka proses

analisis ini dinamakan hirarki [5]. Untuk lebih jelasnya dapat dilihat pada gambar 2.8.



Gambar 2. 8. Hierarki Penyelesaian Metode AHP

2.6.1.2. Pembobotan

Pada Tahap ini dilakukan *comparative judgement*, yaitu membuat penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkat di atasnya. Penilaian ini merupakan inti dari AHP, karena penilaian akan berpengaruh terhadap prioritas elemen – elemen. Penilaian ini ditampilkan dalam bentuk matriks perbandingan berpasangan (*matrix pairwise comparison*) berukuran $n \times n$ dimana n adalah jumlah elemen yang dibandingkan [5]. *Pairwise comparison* diimplementasikan dengan dua tahap:

1. Menentukan secara kualitatif kriteria mana yang lebih penting. misalnya mengurutkan ranking/peringkat.
2. Menggunakan masing-masing kriteria dengan bobot kuantitatif seperti peringkat yang memuaskan.

Proses perbandingan dapat dikemukakan dengan penyusunan skala variabel. Dalam penyusunan skala kepentingan ini didasarkan pada nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty [5] yang ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Skala Perbandingan Saaty

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama penting.
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.
5	Elemen yang satu lebih penting daripada yang lainnya.
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak lebih penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan
Kebalikan	Jika aktivitas I mendapat satu angka dibandingkan dengan aktivitas j, maka j memiliki nilai kebalikan disbanding dengan i.

Pengisian nilai tabel perbandingan berpasangan dilakukan berdasarkan kebijakan pembuat keputusan dengan melihat tingkat kepentingan antar satu elemen dengan elemen yang lainnya. Proses perbandingan berpasangan, dimulai dari perbandingan kriteria misalnya A1, A2 dan A3. Maka susunan elemen-elemen yang dibandingkan tersebut akan tampak seperti pada Tabel 2.4.

Tabel 2. 4. Perbandingan berpasangan

	A1	A2	A3
A1	1		
A2		1	
A3			1

2.6.1.3. Sintesa Prioritas

Dari setiap matrik perbandingan harus dicari prioritas lokal, lalu untuk mendapatkan prioritas global maka harus dilakukan sintesa diantara prioritas lokal pada matriks *pairwise comparison* di setiap tingkat. Langkah– langkah sintesa dalam setiap matrik adalah sebagai berikut [5] :

1. Menjumlahkan nilai dalam tiap kolom matrik sehingga diperoleh jumlah nilai masing – masing kolom.
2. Untuk menormalisasi matrik, dengan cara melakukan pembagian tiap nilai dalam kolom matrik dengan jumlah nilai kolom masing – masing.
3. Menjumlahkan nilai tiap baris pada matrik yang dihasilkan pada langkah b. Nilai ini kemudian dibagi dengan jumlah elemen dalam tiap baris. Nilai yang dihasilkan berupa vektor prioritas atau *eigen vector* yang menunjukkan nilai prioritas menyeluruh untuk tiap elemen.

2.6.1.4. Logikal Konsistensi

Penilaian dalam membandingkan antara satu kriteria dengan kriteria yang lain adalah bebas satu sama lain, dan hal ini dapat mengarah pada ketidak konsistensian. Maka untuk mengetahui tingkat konsistensi penilaian kriteria-kriteria, metode AHP harus dilengkapi dengan penghitungan Indeks Konsistensi (*Consistency Index*) [5]. telah membuktikan bahwa indeks konsistensi dari matrik ber-ordo n dapat diperoleh dengan rumus :

$$CI = \frac{\lambda_{\max} - n}{(n-1)} \quad (2.1)$$

Keterangan Persamaan 2.1 :

CI = Indeks Konsistensi (*Consistency Index*)

λ_{\max} = Nilai *eigen (eigen value)* matriks berordo n

n = Jumlah elemen yang dibandingkan dalam matriks.

Sedangkan *eigen value* (λ_{\max}) didapat dengan menjumlahkan hasil perkalian jumlah kolom pada matrik sebelum dinormalisasi dengan *eigen vector* pada tiap elemen. Semakin dekat nilai λ_{\max} dengan n , maka semakin konsisten hasilnya. Batas ketidak konsistensian dapat di ukur dengan menggunakan rasio konsistensi (CR) yang dapat dirumuskan sebagai berikut :

$$CR = \frac{CI}{RI} \quad (2.2)$$

Keterangan Persamaan 2.2 :

CR = Rasio Konsistensi (*Consistency Ratio*)

CI = Indeks Konsistensi (*Consistency Index*)

RI = Indeks Random Konsistensi

Apabila nilai CR lebih kecil dari 10% atau 0.1, maka ketidak konsistensian penilaian masih dapat diterima, jika tidak maka harus dilakukan penilaian ulang terhadap tiap elemen. Sedangkan daftar Indeks Random Konsistensi (RI) dapat dilihat pada Tabel 2.5.

Tabel 2. 5. Indeks Random Konsistensi

n	2	3	4	5	6	7
RI	0	0.58	0.9	1.12	1.24	1.32

2.6.2. Matriks Perbandingan

Apabila membandingkan suatu kriteria dalam matriks dengan kriteria itu sendiri, misalnya sub kriteria 1 dengan sub kriteria 1, maka pada sisi diagonal perbandingan itu harus bernilai 1. Membandingkan sub kriteria pertama (kolom sebelah kiri) dengan sub kriteria yang kedua (di baris puncak) dan taksir nilai numeriknya. Nilai kebalikannya digunakan untuk membandingkan sub kriteria kedua dengan sub kriteria pertama tadi. Misalnya, jika kedua kriteria itu adalah kertas dan kertas, yang pertama beratnya lima kali lebih berat kertas yang kedua, maka kertas yang kedua beratnya seperlima kali berat kertas yang pertama. Contoh dari matriks perbandingan adalah seperti pada tabel 2.6.

Tabel 2. 6. Tabel matriks perbandingan

Kriteria	Kriteria 1	Kriteria 2	Kriteria 3
Kriteria 1	1		
Kriteria 2		1	
Kriteria 3			1

2.6.3. Nilai Total

Setelah dihitung melaui matriks perbandingan antar alternatif terhadap 1 kriteria maka akan didapat nilai total dari hasil pembobotan yang telah dilakukan pada proses sebelumnya. Contoh dari matriks tabel hasil nilai total adalah seperti pada tabel 2.7.

Tabel 2. 7. Tabel hasil *score*

	Kriteria 1	Kriteria 2	Kriteria 3	Nilai Total
Alternatif 1				
Alternatif 2				
Alternatif 3				

2.7. RBAC (*Role Base Access Control*)

RBAC (Role Based Access Control) adalah sistem yang diterapkan pada aplikasi yang berhubungan dengan pengontrolan akses sumber daya. RBAC memberikan hak akses untuk peran (*roles*). Perancang kebijakan atau administrator sangat berperan dalam memberikan hak kepada para pelaku, sehingga subjek akan mendapatkan akses ke objek melalui *role* yang telah diberikan oleh administrator [7]. Hubungan antara hak akses dengan para pelaku sistem, pertama RBAC memberikan tugas keamanan pada kontrol akses sebagai prioritas tertinggi untuk mengontrol akses ke sumber daya. Hal tersebut mengakibatkan RBAC akan menerapkan keamanan yang sangat ketat dalam melakukan kontrol akses ke sumber daya. Kedua RBAC dalam menerapkan hak akses kepada pengguna membutuhkan waktu yang singkat, dengan cara menghubungkan subjek dengan *role*, sehingga memerlukan penunggasan hak akses untuk *role* pada setiap subjek [7].

Kontrol akses dalam mengambil keputusan ditentukan oleh *role*, sehingga pengguna sebagai bagian dari sebuah organisasi akan mendapatkan hak akses sesuai dengan *role* yang didapatkannya. Kebijakan yang dilakukan oleh RBAC akan membuat control akses yang didapatkan oleh pengguna

berdasarkan keputusan yang diperoleh dalam sebuah organisasi. Pengguna tidak dapat mengambil hak akses pengguna lain, [8]. Dalam RBAC, *role* didefinisikan sebagai suatu gagasan yang merupakan dasar dari kebijakan kontrol akses [8]. Dengan kata lain pendefinisan *role* adalah tugas yang didapatkan oleh seseorang sesuai dengan tanggung jawabnya.

2.8. PHP (*Hypertext Preprocessor*)

PHP adalah singkatan dari *Hypertext Preprocessor* yakni salah satu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari *Personal Home Page Tools*. Selanjutnya diganti FI (*Forms Interpreter*). Sejak versi 3.0, nama bahasa ini diubah menjadi PHP (*Hypertext Preprocessor*) dengan singkatan PHP.

2.9. Framework Yii2

Sebuah *framework* untuk bahasa pemrograman PHP dimana framework ini telah didukung oleh aspek *modularity* dan *eloquent database*. Arsitektur perangkat lunak ini menggunakan *Model-View-Controller*, namun dengan aspek *modularity*, maka dapat dibuat modul dimana setiap modul memiliki MVC tersendiri. Sedangkan *eloquent database* adalah basisdata berupa objek di dalam sebuah bahasa pemrograman PHP.

2.10. MySQL (My Structured Query Language)

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau lebih dikenal dengan DaBagianase Management System. Umumnya MySQL adalah perangkat lunak gratis yang berlisensikan GNU *General Public*

License (GPL). Namun ada sebagian yang mempunyai lisensi komersial. MySQL adalah implementasi dari system manajemen basis data relasional (RDBMS) atau *Relational Data Base Management System*. MySQL merupakan *Inheritance* (Turunan) dari konsep utama dalam basis data yang telah ada sebelumnya yakni SQL (*Structured Query Language*). SQL merupakan sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data yang dikerjakan secara otomatis dan mudah.

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai manajemen basis data, MySQL mendukung operasi basisdata transaksional mau-pun operasi basisdata non-transaksional.

Pada metode operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal sistem kerja dibandingkan perangkat lunak manajemen basis data yang lainnya. Namun demikian pada metode non-transaksional tidak menjamin atas reliabilitas (kehandalan) terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan Tugas Akhir. Pada bagian awal dibahas mengenai analisis permasalahan yang ingin diselesaikan. Selanjutnya dibahas mengenai perancangan program untuk memberikan gambaran umum mengenai sistem yang dibuat. Pendekatan yang digunakan dalam perancangan ini adalah pendekatan rancangan berorientasi objek yang direpresentasikan dengan menggunakan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain: cakupan permasalahan, rencana bisnis, perhitungan ranking hutang, dan perhitungan ranking pelanggan. Berikut pembahasan bagian-bagian tahap analisis.

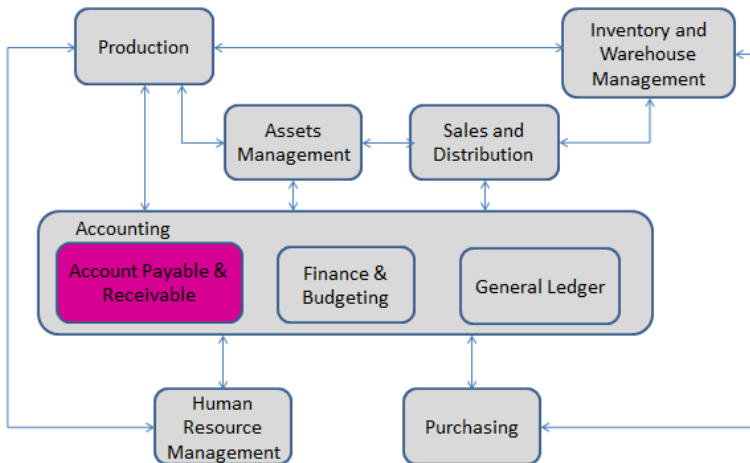
3.1.1. Analisis Proses Bisnis

Setiap ERP memiliki proses bisnis yang berbeda-beda. Pada Tugas Akhir ini, aplikasi ERP yang dikembangkan memiliki proses bisnis tersendiri dan dibandingkan dengan aplikasi ERP lainnya yang cukup banyak digunakan oleh perusahaan-perusahaan, seperti Odoo, Adempiere, dan InoERP. Pada subbab ini akan dijelaskan tentang analisa proses bisnis yang telah ada. Proses bisnis yang dimiliki oleh Odoo, Adempiere, dan InoERP ditunjukkan pada Lampiran A.1, Lampiran A.2, dan Lampiran A.3. Perbandingan masing-masing ERP di tunjukkan pada Tabel 3.1.

Tabel 3. 1. Kekurangan dan kelebihan pada Odoo, Adempiere dan InoERP

Nama Modul	Odoo	Adempiere	InoERP
Account Receivable and Account Payable	Kontra: tidak dapat melakukan pembayaran secara kredit Kontra: tidak dapat melakukan pembayaran secara kredit	-	Pro: Fitur sudah lengkap Kontra: tidak ada analisa transaksi yang pernah dilakukan perusahaan Pro: Fitur sudah lengkap Kontra: tidak ada analisa transaksi yang pernah dilakukan customer

Tabel 3.1 menunjukkan kelebihan dan kekurangan dari masing-masing ERP pada modul *Account Payable* dan *Account Receivable*. Berdasarkan kekurangan yang dimiliki oleh setiap ERP maka perlu dikembangkan sebuah aplikasi ERP yang sesuai dengan kebutuhan pada perencanaan bisnis. Terdapat dua jenis proses bisnis yang digunakan, yaitu *Make-To-Order* (MTO) dan *Make-To-Stock* (MTS). MTO adalah sebuah proses bisnis dimana produksi didasarkan pada permintaan yang dilakukan oleh *sales order*. MTS adalah sebuah proses bisnis dimana produksi didasarkan oleh peramalan penjualan. Proses bisnis yang dirancang sesuai dengan kebutuhan *business plan* dimana proses bisnis ini telah mengakomodasi kekurangan Odoo, Adempiere, dan InoERP ditunjukkan pada Gambar 3.1, Gambar 3.2, Gambar 3.3, Gambar 3.4, Gambar 3.5, Gambar 3.6, dan Gambar 3.7.



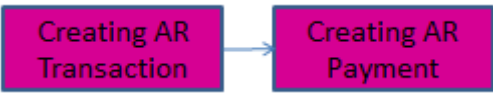
Gambar 3. 1. Proses Bisnis ERP 2016 Level 0

Pada Gambar 3.1, modul-modul keuangan (AP/AR, *Finance*, dan *General Ledger*) merupakan sentral dari sistem ERP. Sedangkan, modul *Sales and Distribution* berinteraksi dengan modul *Inventory and Warehouse Management* dan modul *Assets Management*. Modul *Production* berinteraksi dengan modul *Asset Management*, *Finance* dan *Warehouse Management*. Modul *Human Resource* berinteraksi dengan modul AP/AR dan *Production*. Sedangkan modul *Purchasing* berinteraksi dengan modul *Warehouse Management* dan AP/AR. Pada Tugas Akhir ini, lebih rinci akan dibahas proses bisnis modul yang berwarna latar belakang ungu, yaitu modul *Account Payable* dan *Account Receivable*. Modul *Account Payable* dan *Account Receivable* secara umum dijelaskan pada deskripsi umum sistem sub-bab 3.2.1. dan sub-bab 3.2.2.

Proses bisnis ERP level 0 dapat dipecah kembali menjadi proses-proses bisnis yang lebih rinci. Proses bisnis secara keseluruhan ditunjukkan pada Lampiran A.4. Hasil pemecahan proses bisnis untuk modul *Account Payable* dan *Account Receivable* ditunjukkan pada Gambar 3.2 dan Gambar 3.3.



Gambar 3. 2. Proses Bisnis *Level 1 Account Payable*



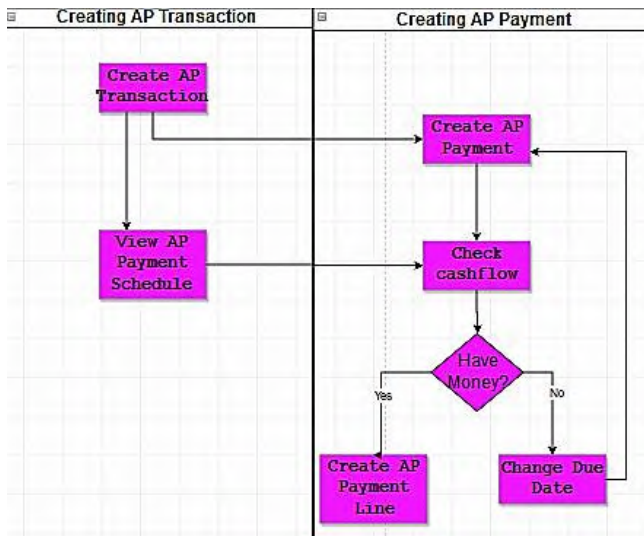
Gambar 3. 3. Proses Bisnis *Level 1 Account Receivable*

Pada Gambar 3.2 dan Gambar 3.3 ditunjukkan proses bisnis *level 1 Account Payable* dan proses bisnis *level 1 Account Receivable*. Penjelasan proses bisnis *level 1* terdapat pada Tabel 3.2.

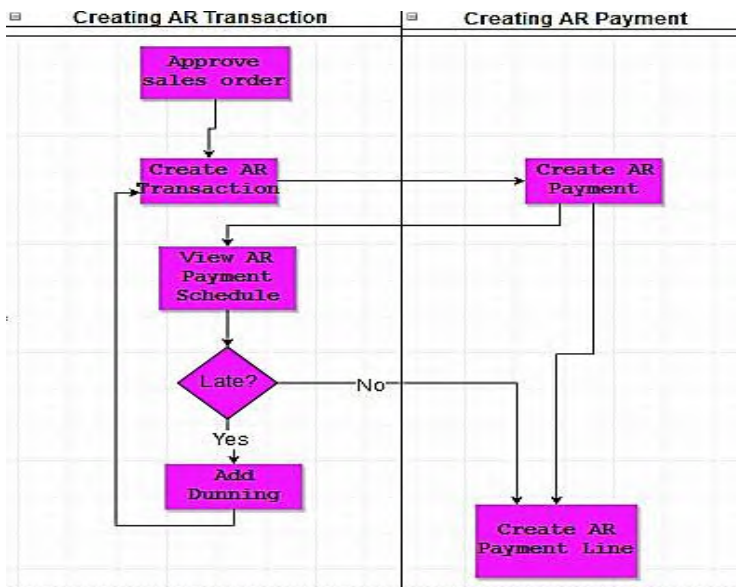
Tabel 3.1. Deskripsi Proses Bisnis *Level 1*

No	Proses Bisnis	Keterangan
1	<i>Creating AP Transaction</i>	Merupakan proses bisnis untuk membuat transaksi hutang perusahaan.
2	<i>Creating AP Payment</i>	Merupakan proses bisnis untuk membuat pembayaran hutang perusahaan.
3	<i>Creating AR Transaction</i>	Merupakan proses bisnis untuk membuat transaksi piutang perusahaan..
4	<i>Creating AR Payment</i>	Merupakan proses bisnis untuk membuat pembayaran piutang perusahaan.

Dari masing-masing proses bisnis *level 1* dapat dipecah kembali menjadi proses bisnis yang lebih rinci lagi. Proses bisnis ditunjukkan pada Gambar 3.4 dan Gambar 3.5.



Gambar 3. 4. Proses Bisnis *Level 2 Account Payable*



Gambar 3. 5. Proses Bisnis *Level 2 Account Receivable*

Pada Gambar 3.4 ditunjukkan proses bisnis *level 2 Account Payable* dan Gambar 3.5 ditunjukkan proses bisnis *level 2 Account Receivable*. Penjelasan proses bisnis *level 2* terdapat pada Tabel 3.2. dan Tabel 3.3.

Tabel 3. 2. Deskripsi Proses Bisnis *Account Payable Level 2*

No	Proses Bisnis	Keterangan
1	<i>Create AP Transaction</i>	Merupakan proses bisnis untuk membuat transaksi hutang.
2	<i>View AP Transaction</i>	Merupakan proses bisnis untuk melihat daftar transaksi hutang.
3	<i>Create AP Payment</i>	Merupakan proses bisnis untuk membuat pembayaran hutang
4	<i>Check Cashflow</i>	Merupakan proses bisnis untuk mengecek uang kas perusahaan.
5	<i>Change Due Date</i>	Merupakan proses bisnis untuk mengubah tanggal jatuh tempo.

Tabel 3. 3. Deskripsi Proses Bisnis *Account Receivable Level 2*

No	Proses Bisnis	Keterangan
1	<i>Create AR Transaction</i>	Merupakan proses bisnis untuk membuat transaksi piutang.
2	<i>View AR Transaction</i>	Merupakan proses bisnis untuk melihat daftar transaksi piutang.
3	<i>Create AR Payment</i>	Merupakan proses bisnis untuk membuat pembayaran piutang.

No	Proses Bisnis	Keterangan
4	<i>Add Dunning</i>	Merupakan proses bisnis untuk mencatat keterlambatan pembayaran pelanggan.
5	<i>Create AR Payment Line</i>	Merupakan proses bisnis untuk membuat pembayaran piutang pelanggan secara kredit.

Dari masing-masing proses bisnis *level 2* dapat dipecah kembali menjadi proses bisnis yang lebih rinci lagi. Proses bisnis ditunjukkan pada Gambar 3.7 dan Gambar 3.8. Gambar 3.7 merupakan proses bisnis *level 3 Account Payable* dan Gambar 3.8 merupakan proses bisnis *level 3 Account Receivable*. Pada proses bisnis *level 3* menjelaskan detail dari kelas, fungsi, dan halaman yang ditampilkan untuk melakukan proses bisnis yang terdapat pada *level 2*.

3.1.2. Analisis Data

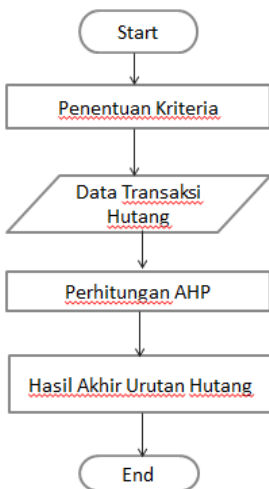
Konsep autentikasi dan otorisasi data yang digunakan adalah *multitenancy separated database, separated schema* yang dijelaskan pada subbab 2.4. Pada konsep ini, data milik setiap *tenant* akan disimpan di basis data yang terpisah. Sehingga tidak ada 2 atau lebih *tenant* menggunakan tabel dan basis data yang sama. Penambahan *tenant* baru akan menyebabkan penambahan sebuah basis data baru. Penggunaan konsep *separated database, separated schema* dimaksudkan agar data milik setiap *tenant* dapat terjamin kerahasiaannya. Keuntungan lainnya adalah kerumitan proses *maintenance* basis data dapat dikurangi.

Konsep basis data terdistribusi yang diterapkan adalah replikasi yang dijelaskan pada subbab 2.3. Replikasi memungkinkan penyalinan setiap tabel basis data ke *node-node* penyusun basis data terdistribusi. Kegagalan sebuah *node* tidak akan menyebabkan basis data berhenti bekerja. Sebaliknya, hal

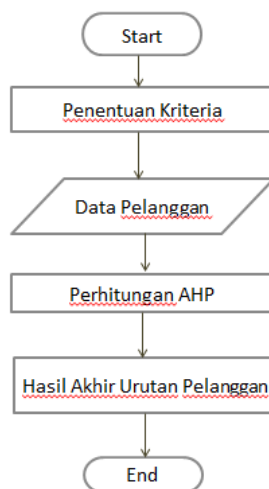
tersebut akan memicu mekanisme sinkronisasi jika *node* yang mati kembali hidup.

3.1.3. Analisis Metode Perangkingan

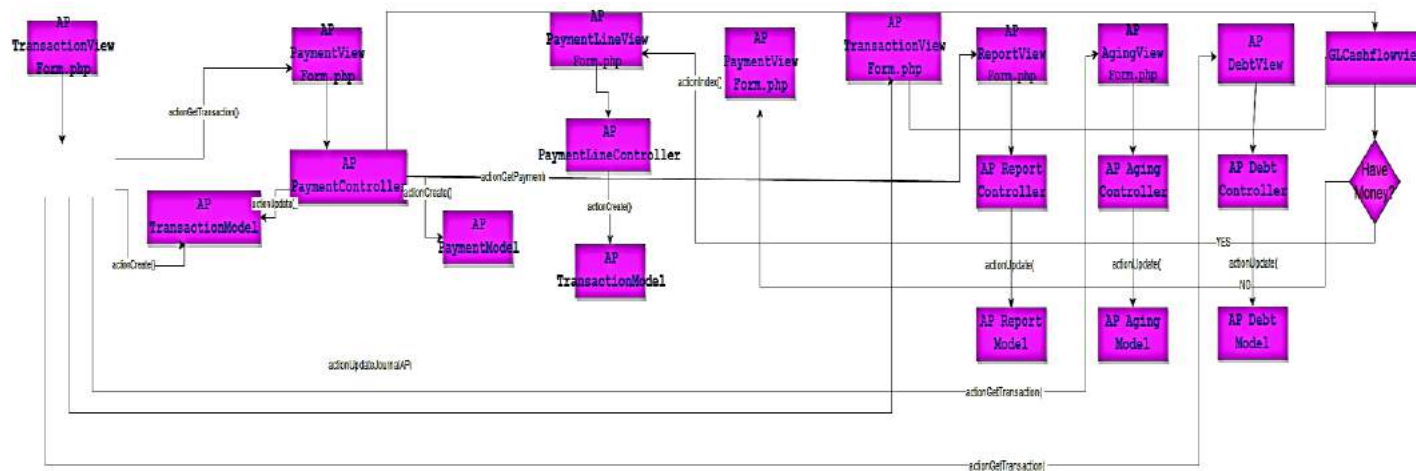
Sistem yang dibangun merupakan sistem perangkingan hutang dan pelanggan yang digunakan untuk membantu *user* (manajer keuangan) dalam mencari rekomendasi hutang mana yang sebaiknya dibayar terlebih dahulu oleh perusahaan. Dan merekomendasikan pelanggan terbaik Untuk membangun sistem ini diperlukan data-data pendukung yaitu data hutang perusahaan yang belum lunas, data transaksi pembelian pelanggan dimana semua data pendukung tersebut disimpan ke dalam *database*. Hasil akhir dari sistem ini berupa rekomendasi hutang yang sebaiknya dibayar terlebih dahulu dan rekomendasi pelanggan terbaik dengan beberapa kriteria sebagai pertimbangan dengan menggunakan metode yang telah dibahas pada subbab 2.6 yaitu metode AHP (*Analytical Hierarchy Process*). Gambaran umum sistem tersebut dijelaskan pada Gambar 3.8 dan Gambar 3.9.



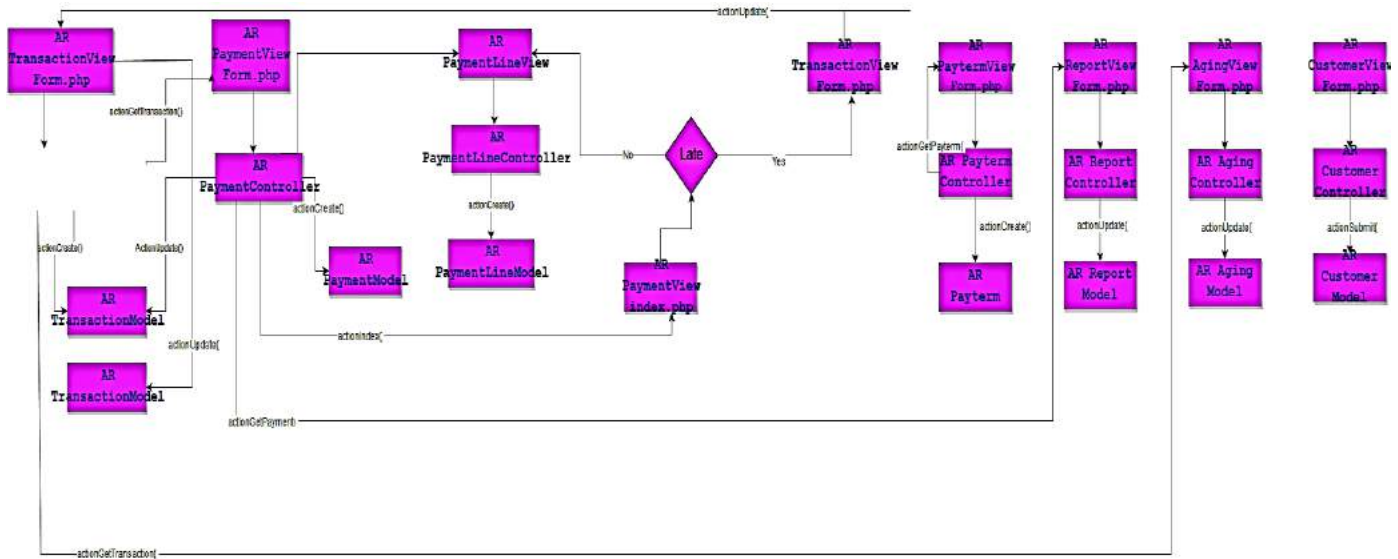
Gambar 3. 6. Alur Kerja Sistem Perangkingan Hutang (1)



Gambar 3. 7. Alur kerja istem Perangkingan Pelanggan (2)



Gambar 3. 8. Proses Bisnis *Level 3 Account Payable*



Gambar 3. 9. Proses Bisnis Level 3 Account Receivable

3.1.3.1. Penentuan Kriteria Perangkingan Hutang

Berdasarkan hasil penelitian terkait tentang teknik mengelola hutang perusahaan, didapat 6 kriteria yang dapat digunakan dalam perangkingan hutang perusahaan menurut tingkat kepentingan hutang tersebut, yaitu tanggal toleransi keterlambatan yang diberikan supplier (*payment tolerance*), sisa waktu pembayaran sebelum jatuh tempo (*time left*), diskon pelunasan sebelum jatuh tempo (*discount early payment*) [9], denda keterlambatan (*fine per month*) [9], frekuensi keterlambatan (*late frequency*) [9], dan jumlah hutang (*total debt*) [10]. Berikut penjelasan kriteria-kriteria tersebut:

1. *Payment Tolerance (days)*

Setiap *supplier* memiliki ketentuan tentang toleransi keterlambatan pembayaran yang berbeda-beda. Ada *supplier* yang memperbolehkan pelanggan mereka untuk melunasi hutang meskipun telah melewati tanggal jatuh tempo, dan ada pula yang mengharuskan pelanggan mereka untuk membayar tepat waktu dan mungkin tidak akan memperbolehkan pelanggan tersebut untuk melakukan hutang dagang kembali apabila pernah melewati tanggal jatuh tempo. Sehingga kriteria ini dapat dijadikan pertimbangan untuk meranking hutang perusahaan.

2. *Time Left (days)*

Hutang Dagang memiliki tanggal jatuh tempo, dimana hutang tersebut harus sudah terbanyak pada tanggal jatuh tempo yang telah ditentukan sebelumnya. Semakin mendekati tanggal jatuh tempo, dapat diasumsikan hutang tersebut sangat penting untuk dibayar terlebih dahulu, dan sebaliknya, sehingga kriteria ini dapat dijadikan salah satu pertimbangan untuk meranking hutang perusahaan karena apabila perusahaan terlambat membayar hutang kepada *supplier* maka akan menurunkan kepercayaan *supplier*.

3. *Discount Early Payment (IDR)*

Potongan harga yang diberikan oleh *supplier* kepada perusahaan apabila perusahaan melunasi hutang dagang lebih awal, sebelum tanggal jatuh tempo.

4. *Fine per Month (IDR)*

Denda atau tagihan tambahan yang harus dibayarkan kepada *supplier* apabila perusahaan belum melunasi hutang dagang setelah lewat tanggal jatuh tempo.

5. *Late Frequency (%)*

Ketika perusahaan pernah melakukan pembayaran melewati jatuh tempo kepada *supplier*, hal ini menjadi perhatian khusus perusahaan untuk tidak mengulangi keterlambatan pembayaran tersebut kembali, karena akan menurunkan kepercayaan *supplier* kepada perusahaan sehingga perusahaan akan masuk daftar *blacklist* pada *supplier* tersebut.

6. *Total Debt (IDR)*

Jumlah hutang yang harus dibayarkan perusahaan menjadi pertimbangan dalam sistem perankingan karena harus mempertimbangkan dengan arus kas perusahaan pada saat itu.

3.1.3.2. Perhitungan Metode AHP (*Analytical Hierarchy Process*) Perangkingan Hutang

Langkah pertama dalam perhitungan menggunakan metode AHP yaitu dengan melakukan permbobotan antar kriteria. Nilai pembobotan yang digunakan berdasarkan penggalan informasi berupa form kuesioner yang terdapat pada Lampiran B.1. Pengisian kuesioner dilakukan oleh para pelaku yang dinilai cukup memiliki pengetahuan tentang akuntansi, antara lain mahasiswa/mahasiswi tingkat akhir jurusan Akuntansi, Pegawai BUMN dan Swasta yang bekerja di bidang akuntansi. Hasil

kuesioner penentuan kriteria terdapat pada Lampiran B.2. Secara ringkas hasil kuesioner dapat dilihat pada tabel 3.4.

Tabel 3. 4. Hasil Kuesioner

Kriteria	Kepentingan			
	Tidak Berpengaruh	Sedikit Berpengaruh	Berpengaruh	Sangat Berpengaruh
Fine Per Month (IDR)	0	1	5	4
Total Debt (IDR)	1	2	3	4
Payment Tolerance (Days)	0	3	3	4
Time Left (Days)	0	2	4	4
Late Frequency (%)	0	1	5	4
Discount Early Payment (IDR)	1	1	4	4

Dari 10 orang yang mengisi kuesioner, pada kriteria *Fine per Month* terdapat 1 orang menilai sedikit berpengaruh, 5 orang menilai berpengaruh, dan 4 orang menilai sangat berpengaruh. Kemudian pada kriteria *Total Debt*, terdapat 1 orang menilai tidak berpengaruh, 2 orang menilai sedikit berpengaruh, 3 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Pada kriteria *Payment Tolerance*, terdapat 3 orang menilai sedikit berpengaruh, 3 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Kemudian pada kriteria *Time Left*, terdapat 2 orang menilai sedikit berpengaruh, 4 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Pada kriteria *Late Frequency*, terdapat 1 orang menilai sedikit berpengaruh, 5 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Kemudian pada kriteria *Discount Early Payment*, terdapat 1 orang menilai tidak berpengaruh, 1 orang menilai sedikit berpengaruh, 4

orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh.

Dari hasil kuesioner, berdasarkan referensi pengolahan hasil kuesioner [11] diasumsikan untuk penilaian tidak berpengaruh bernilai 1, sedikit berpengaruh bernilai 2, berpengaruh bernilai 3 dan sangat berpengaruh bernilai 4. Maka dapat dihitung nilai total dan bobot dari masing-masing kriteria seperti yang ditunjukkan pada Tabel 3.5.

Tabel 3. 5. Pembobotan kuesioner

Kriteria	Kepentingan				Nilai Total	Urutan	Bobot
	Tidak Berpengaruh	Sedikit Berpengaruh	Berpengaruh	Sangat Berpengaruh			
Fine Per Month (IDR)	0	2	15	16	33	2	5
Total Debt (IDR)	1	4	9	16	30	6	1
Payment Tolerance (Days)	0	6	9	16	31	4	3
Time Left (Days)	0	4	12	16	32	3	4
Late Frequency (%)	0	2	15	16	33	1	6
Discount Early Payment (IDR)	1	2	12	16	31	5	2

.Urutan kriteria ditentukan berdasarkan nilai total tertinggi. Apabila terdapat nilai yang sama, dilakukan perbandingan urutan

yang dilakukan oleh *responden*. Pembobotan kriteria dilakukan dengan melakukan perbandingan berpasangan antar kriteria dengan membandingkan baris dan kolom masing-masing kriteria. Pembobotan kriteria tersebut ditunjukkan pada Tabel 3.6 dan hasil perbandingannya ditunjukkan pada Tabel 3.7.

Tabel 3. 6. Pembobotan kriteria

Kriteria	Fine Per Month (IDR)	Total Debt (IDR)	Payment Tolerance (Days)	Time Left (Days)	Late Frequency (%)	Discount Early Payment (IDR)
Fine Per Month (IDR)	5/5	5/1	5/3	5/4	5/6	5/2
Total Debt (IDR)	1/5	1/1	1/3	1/4	1/6	1/2
Payment Tolerance (Days)	3/5	3/1	3/3	3/4	3/6	3/2
Time Left (Days)	4/5	4/1	4/3	4/4	4/6	4/2
Late Frequency (%)	6/5	6/1	6/3	6/4	6/6	6/2
Discount Early Payment (IDR)	2/5	2/1	2/3	2/4	2/6	2/2

Tabel 3. 7. Hasil perhitungan bobot kriteria

Kriteria	Fine Per Month (IDR)	Total Debt (IDR)	Payment Tolerance (Days)	Time Left (Days)	Late Frequency (%)	Discount Early Payment (IDR)
Fine Per Month	1	5	1.667	1.25	0.833	2.5

Kriteria	Fine Per Month (IDR)	Total Debt (IDR)	Payment Tolerance (Days)	Time Left (Days)	Late Frequency (%)	Discount Early Payment (IDR)
(IDR)						
Total Debt (IDR)	0.2	1	0.333	0.25	0.167	0.5
Payment Tolerance (Days)	0.6	3	1	0.75	0.75	1.5
Time Left (Days)	0.8	4	1.333	1	0.667	2
Late Frequency (%)	1.2	6	2	1.5	1	3
Discount Early Payment (IDR)	0.4	2	0.667	0.5	0.333	1
Total	4.2	21	7	5.25	3.75	10.5

Selanjutnya menghitung nilai prioritas kriteria (*priority vector*) dengan rumus jumlah kolom (Total) dibagi dengan jumlah kriteria. Perhitungan nilai prioritas kriteria ditunjukkan pada Tabel 3.8.

Tabel 3. 8 Nilai prioritas kriteria (Priority vector)

Kriteria	Prioritas Kriteria
Fine Per Month (IDR)	0.111
Total Debt	0.100
Payment Tolerance(Days)	0.301
Time Left (Days)	0.259
Late Frequency	0.043
Discount Early Payment	0.186

Kemudian menghitung nilai CR (*Consistensy Ratio*) untuk menentukan penilaian perbandingan kriteria tersebut bersifat konsisten atau tidak. Ada 3 tahapan yang harus dilakukan, yaitu :

1. Menentukan nilai eigen maksimum (λ_{maks})

λ maks diperoleh dengan menjumlahkan hasil perkalian jumlah matriks kolom dengan *priority vector* untuk setiap kriteria. Untuk jumlah matriks kolom telah ditunjukkan pada tabel 3.8, dan nilai *priority vector* telah ditunjukkan pada tabel 3.9. Sehingga didapat nilai $\lambda_{\text{maks}} = 6.108$.

2. Menghitung Indeks Konsistensi (CI)

CI diperoleh dengan menghitung λ_{maks} dikurangi jumlah kriteria, kemudian dibagi dengan jumlah kriteria dikurangi 1. Sehingga didapat nilai CI = 0.02

3. Menghitung rasio konsistensi (CR)

CR diperoleh dengan membagi nilai CI dengan nilai RI. RI didapat dari daftar Indeks random konsistensi pada tabel 3.9. Untuk $n = 6$ adalah 1,24 dimana n adalah jumlah kriteria. Sehingga didapat nilai CR = 0.071

Tabel 3. 9. Random Index Saaty

n	RI
2	0
3	0.58
4	0.9
5	1.12
6	1.24
7	1.32

Dari hasil perhitungan dapat diketahui bahwa nilai CR = 0.071, karena $CR < 0,1$, artinya preferensi pembobotan adalah konsisten. Apabila nilai $CR > 0,1$ maka preferensi pembobotan

tidak konsisten dan harus dilakukan pembobotan kembali hingga nilai $CR < 0,1$.

Setelah mendapatkan konsistensi dari hasil kriteria, selanjutnya menyusun alternatif-alternatif yang telah ditentukan dalam bentuk matriks berpasangan untuk masing-masing kriteria. sehingga akan ada sebanyak n buah matriks berpasangan alternatif dimana nilai n adalah jumlah kriteria yang ada. Berikut merupakan data transaksi yang akan dijadikan alternatif yang ditunjukkan pada Tabel 3.10.

Tabel 3. 10. Data transaksi

Alternatif / Kriteria	Fine Per Month (IDR)	Total Debt (IDR)	Payment Tolerance (Days)	Time Left (Days)	Late Frequency (%)	Discount Early Payment (IDR)
APTRANS003	22649999.75	45299995	9	6	1	28900000
APTRANS004	150323855.3	300000000	13	6	1	107374182.4
APTRANS005	24000000	60000000	11	3	0	14120000

Setelah diketahui nilai-nilai kriteria untuk setiap alternatif, selanjutnya dibandingkan setiap alternatif untuk setiap kriteria. Berikut penjelasan untuk setiap kriteria yang ada:

1. *Fine Per Month (IDR)*

Pada kriteria denda setiap bulan, nilai matriks didapat dengan melakukan perbandingan baris dibanding kolom. Jika hasil denda yang ditentukan supplier kurang dari atau sama dengan nol maka nilai kriteria denda tersebut akan didefinisikan sebesar 0.001. Matriks perbandingan bebasangan terhadap kriteria pertama ditunjukkan pada tabel 3.11 dan hasil normalisasinya ditunjukkan pada Tabel 3.12

Tabel 3. 11. *Pairwise comparison kriteria 1*

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	6.636	1.059
APTRANS0004	0.150	1	0.159
APTRANS0005	0.943	6.263	1

Tabel 3. 12. Normalisasi kriteria 1

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.477	0.477	0.477
APTRANS0004	0.072	0.072	0.072
APTRANS0005	0.451	0.41	0.451

2. *Total Debt (IDR)*

Pada kriteria jumlah hutang perusahaan, nilai matriks didapat dengan melakukan perbandingan baris dibanding kolom. Matriks perbandingan bebasangan terhadap kriteria kedua ditunjukkan pada Tabel 3.13 dan hasil normalisasinya ditunjukkan pada Tabel 3.14

Tabel 3. 13. *Pairwise comparison kriteria 2*

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	6.623	1.325
APTRANS0004	0.151	1	0.2
APTRANS0005	0.755	5	1

Tabel 3. 14. Normalisasi kriteria 2

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.525	0.525	0.525
APTRANS0004	0.079	0.079	0.079
APTRANS0005	0.396	0.396	0.396

3. *Payment Tolerance (days)*

Pada kriteria toleransi keterlambatan, perusahaan menghendaki *supplier* dengan toleransi pembayaran paling lama . Apabila *supplier* tidak memberikan toleransi akan keterlambatan pembayaran, maka nilai kriteria *payment*

tolerance akan didefinisikan sebesar 0.001. Matriks perbandingan bebasangan terhadap kriteria ketiga ditunjukkan pada Tabel 3.15 dan hasil normalisasinya ditunjukkan pada Tabel 3.16

Tabel 3. 15. *Pairwise comparison* kriteria 3

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	0.692	0.818
APTRANS0004	1.444	1	1.189
APTRANS0005	1.222	0.846	1

Tabel 3. 16. Normalisasi kriteria 3

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.273	0.273	0.273
APTRANS0004	0.394	0.394	0.394
APTRANS0005	0.333	0.333	0.333

4. *Time Left (days)*

Pada kriteria sisa waktu pembayaran (*time left*), hutang yang memiliki sisa waktu paling dekat adalah yang paling penting untuk dibayar terlebih dahulu. Matriks perbandingan bebasangan terhadap kriteria keempat ditunjukkan pada tabel 3.17 dan hasil normalisasinya ditunjukkan pada tabel 3.18

Tabel 3. 17. *Pairwise comparison* kriteria 4

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	1	2
APTRANS0004	1	1	2
APTRANS0005	0.5	0.5	1

Tabel 3. 18. Normalisasi kriteria 4

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.4	0.4	0.4
APTRANS0004	0.4	0.4	0.4
APTRANS0005	0.2	0.2	0.2

5. *Frequency of Late (%)*

Kriteria frekuensi keterlambatan menjadi hal yang harus diperhatikan dalam memprioritaskan hutang, semakin sering perusahaan membayar hutang melewati tanggal jatuh tempo, maka akan berdampak buruk bagi perusahaan karena dapat menurunkan kepercayaan *supplier*. Apabila perusahaan belum pernah terlambat dalam membayar hutang terhadap *supplier* tertentu, maka nilai kriteria *frequency of late* didefinisikan sebesar 0.001. Matriks perbandingan bebasangan terhadap kriteria kelima ditunjukkan pada tabel 3.19 dan hasil normalisasinya ditunjukkan pada Tabel 3.20

Tabel 3. 19. *Pairwise comparison* kriteria 5

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	1	0
APTRANS0004	1	1	0
APTRANS0005	1000	1000	0

Tabel 3. 20. Normalisasi kriteria 6

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.001	0.001	0
APTRANS0004	0.001	0.001	0
APTRANS0005	0.998	0.998	0

6. *Discount For Early Payment (IDR)*

Pada kriteria diskon untuk pelunasan sebelum jatuh tempo (*discount for early payment*), perusahaan menginginkan diskon terbesar, sehingga akan lebih baik untuk melunasi hutang sebelum tanggal jatuh tempo. Apabila *supplier* tidak memberikan diskon, maka nilai kriteria ini didefinisikan sebesar 0.001. Matriks perbandingan bebasangan terhadap kriteria keenam ditunjukkan pada tabel 3.21 dan hasil normalisasinya ditunjukkan pada Tabel 3.22

Tabel 3. 21. *Pairwise comparison* kriteria 6

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	1	0.269	2.047

APTRANS0004	3.715	1	7.604
APTRANS0005	0.488	0.131	1

Tabel 3. 22. Normalisasi kriteria 6

	APTRANS0003	APTRANS0004	APTRANS0005
APTRANS0003	0.192	0.192	0.192
APTRANS0004	0.714	0.714	0.714
APTRANS0005	0.094	0.094	0.094

Setelah mendapatkan *priority vector* setiap kriteria untuk ketiga alternatif tersebut, Langkah terakhir adalah menghitung prioritas global/skor untuk ketiga alternatif tersebut sebagai nilai yang digunakan oleh pengambil keputusan. Alternatif diurutkan berdasarkan skor yang terendah karena hutang tersebut akan berdampak paling buruk bagi perusahaan. Hasil skor ditunjukkan pada Tabel 3.23.

Tabel 3. 23. Tota Skor

	APTRANS0003	APTRANS0004	APTRANS0005
Fine Per Month (IDR)	0.477	0.072	0.451
Total Debt (IDR)	0.525	0.079	0.396
Payment Tolerance (Days)	0.273	0.394	0.333
Time Left (Days)	0.4	0.4	0.2
Late Frequency (%)	0.001	0.001	0.665
Discount Early Payment (IDR)	0.192	0.714	0.094
Total	0.327	0.371	0.288

Rumus prioritas global adalah dengan menjumlahkan perkalian dari *priority vector* setiap kriteria pada alternatif dengan jumlah baris kriteria yang sudah dihitung sebelumnya. Dapat kita lihat pada tabel 3.34 rincian *priority vector* setiap alternatif dan kriteria, serta jumlah baris untuk tiap kriteria dan hasil prioritas global untuk ketiga alternatif. Rincian perhitungan untuk prioritas global sebagai berikut :

- Prioritas global APTRANS003 = 0.327
- Prioritas global APTRANS004 = 0.371
- Prioritas global APTRANS005 = 0.288

Dari hasil keseluruhan perhitungan, dapat kita simpulkan bahwa APTRANS005 mempunyai nilai prioritas global yang paling rendah, yaitu 0.288 dan berada di posisi pertama sebagai prioritas hutang yang sebaiknya dibayar terlebih dahulu. APTRANS003 dengan nilai prioritas global sebesar 0.327 dan yang terakhir adalah APTRANS004 dengan nilai prioritas global sebesar 0.371.

3.1.3.3. Penentuan Kriteria Perangkingan Pelanggan

Berdasarkan hasil penelitian terkait tentang *customer relationship management*, didapat 6 kriteria yang dapat digunakan dalam perangkingan pelanggan menurut tingkat loyalitas pelanggan tersebut, yaitu frekuensi keterlambatan (*late frequency*) [12], piutang tak tertagih (*bad debts*) [13], pelunasan sebelum jatuh tempo (*paid before maturity*) [12], total pembayaran (*total payment*) [14], rata-rata pembayaran lunas (*payment complete average*) [13], frekuensi order (*order frequency*) [12]. Berikut penjelasan kriteria-kriteria tersebut:

1. Late Frequency (%)

Presentase pelanggan apabila melakukan pelunasan hutang melewati jatuh tempo yang telah ditentukan sebelumnya. Perusahaan akan lebih berhati-hari untuk memberikan kredit

kembali kepada pelanggan, apabila pelanggan tersebut dinilai tidak disiplin dalam melakukan pelunasan hutang.

2. *Bad Debts (IDR)*

Jumlah hutang pelanggan yang belum terbayarkan sampai beberapa kali perpanjangan toleransi pembayaran. Jumlah sisa hutang tersebut oleh perusahaan masuk dalam jurnal piutang tak tertagih. Apabila pelanggan tercatat memiliki piutang tak tertagih pada perusahaan, maka perusahaan akan mengelompokkan pelanggan tersebut dalam daftar *blacklist*.

3. *Paid Before Maturity (%)*

Presentase pelanggan pernah melunasi hutang/kredit sebelum tanggal jatuh tempo. Hal ini akan menambah penilaian positif perusahaan terhadap pelanggan tersebut sehingga perusahaan akan memberikan perhatian khusus tentang pelayanan kepada pelanggan tersebut.

4. *Total Payment (IDR)*

Jumlah hutang yang telah dibayarkan oleh pelanggan dari keseluruhan transaksi pembelian yang telah dilakukan pelanggan. Semakin banyak pembelian dan pembayaran pelanggan yang telah dilakukan, maka perusahaan memprioritaskan pelanggan tersebut.

5. *Payment Complete Average(Month)*

Rata-rata selang waktu pembayaran pelanggan dari tanggal transaksi dibuat sampai transaksi tersebut berstatus lunas.

6. *Order Frequency*

Frekuensi pelanggan melakukan transaksi pembelian kepada perusahaan. Semakin sering pelanggan tersebut melakukan transaksi, maka perusahaan akan menilai pelanggan tersebut sebagai pelanggan yang loyal.

3.1.3.4. Perhitungan AHP Perangkingan Pelanggan

Langkah pertama dalam perhitungan menggunakan metode AHP yaitu dengan melakukan permbobotan antar kriteria. Nilai pembobotan yang digunakan berdasarkan penggalian informasi berupa form kuesioner yang tedapat pada **Lampiran B.1**. Pengisian kuesioner dilakukan oleh para pelaku yang dinilai cukup memiliki pengetahuan tentang akuntansi, antara lain mahasiswa/mahasiswi tingkat akhir jurusan Akuntansi, Pegawai BUMN dan Swasta yang bekerja di bidang akuntansi.

Hasil kuesioner penentuan kriteria *customer* terbaik dapat dilihat pada Tabel 3.24.

Tabel 3. 24. Hasil Kuesioner

Kriteria	Kepentingan			
	Tidak Berpengaruh	Sedikit Berpengaruh	Berpengaruh	Sangat Berpengaruh
Late Frequency (%)	0	1	3	6
Bad Debts (IDR)	0	2	2	6
Paid Before Maturity (%)	1	1	4	4
Total Payment (IDR)	0	3	3	4
Payment Complete Avg (month)	0	1	8	1
Order Frequency	0	3	3	4

Dari 10 orang yang mengisi kuesioner, pada kriteria *Late Frequency* terdapat 1 orang menilai sedikit berpengaruh, 3 orang menilai berpengaruh, dan 6 orang menilai sangat berpengaruh. Kemudian pada kriteria *Bad Debts*, terdapat 2 orang menilai sedikit berpengaruh, 2 orang menilai berpengaruh dan 6 orang menilai sangat berpengaruh. Pada kriteria *Paid Before*

Maturity, terdapat 1 orang menilai tidak berpengaruh, 1 orang menilai sedikit berpengaruh, 4 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Kemudian pada kriteria *Total Payment*, terdapat 3 orang menilai sedikit berpengaruh, 3 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh. Pada kriteria *Payment Complete Avg*, terdapat 1 orang menilai sedikit berpengaruh, 8 orang menilai berpengaruh dan 1 orang menilai sangat berpengaruh. Kemudian pada kriteria *Order Frequency*, terdapat 3 orang menilai sedikit berpengaruh, 3 orang menilai berpengaruh dan 4 orang menilai sangat berpengaruh.

Dari hasil kuesioner, berdasarkan referensi pengolahan hasil kuesioner [11] dimisalkan untuk penilaian tidak berpengaruh bernilai 1, sedikit berpengaruh bernilai 2, berpengaruh bernilai 3 dan sangat berpengaruh bernilai 4. Maka dapat dihitung nilai total dan bobot dari masing-masing kriteria seperti yang ditunjukkan pada Tabel 3.25.

Tabel 3. 25. Pembobotan Kuesioner

Kriteria	Kepentingan				Nilai Total	Urutan	Bobot
	Tidak Berpengaruh	Sedikit Berpengaruh	Berpengaruh	Sangat Berpengaruh			
Late Frequency (%)	0	2	15	16	35	1	6
Bad Debts (IDR)	1	4	9	16	34	2	5
Paid Before Maturity (%)	0	6	9	16	31	5	2
Total Payment (IDR)	0	4	12	16	31	4	3
Payment Complete Avg (month	0	2	15	16	30	6	1

Order Frequency	1	2	12	16	31	3	4
--------------------	---	---	----	----	----	---	---

. Urutan kriteria ditentukan berdasarkan nilai total tertinggi. Apabila terdapat nilai yang sama, dilakukan perbandingan urutan yang dilakukan oleh *responden*. Pembobotan kriteria dilakukan dengan melakukan perbandingan berpasangan antar kriteria dengan membandingkan baris dan kolom masing-masing kriteria. Pembobotan kriteria tersebut ditunjukkan pada tabel 3.7 dan hasil perbandingannya ditunjukkan pada Tabel 3.26

Tabel 3. 26. Pembobotan kriteria

Kriteria	Late Frequency (%)	Bad Debts (IDR)	Paid Before Maturity (%)	Total Payment (IDR)	Payment Avg (month)	Order Frequency
Late Frequency (%)	6/6	6/5	6/2	6/3	6/1	6/4
Bad Debts (IDR)	5/6	5/5	5/2	5/3	5/1	5/4
Paid Before Maturity (%)	2/6	2/5	2/2	2/3	2/1	2/4
Total Payment (IDR)	3/6	3/5	3/2	3/3	3/1	3/4
Payment Complete Avg (month)	1/6	1/5	1/2	1/3	1/1	1/4
Order Frequency	4/6	4/5	4/2	4/3	4/1	4/4

Dari matriks berpasangan diatas, selanjutnya dapat dihitung bobot prioritas untuk setiap kriteria, dan kemudian dijumlahkan matriks kolomnya seperti pada Tabel 3.27.

Tabel 3. 27. Hasil Pembobotan Kriteria

Kriteria	Late Frequency (%)	Bad Debts (IDR)	Paid Before Maturity (%)	Total Payment (IDR)	Payment Avg (month)	Order Frequency
Late Frequency (%)	1	1.2	3	2	6	1.5
Bad Debts (IDR)	0.833	1	2.5	1.667	5	1.25
Paid Before Maturity (%)	0.333	0.4	1	0.667	2	0.5
Total Payment (IDR)	0.5	0.6	1.5	1	3	0.75
Payment Complete Avg (month)	0.167	0.2	0.5	0.333	1	0.25
Order Frequency	0.667	0.8	2	1.333	4	1
Total	3.5	4.2	10.5	7	21	5.25

Selanjutnya menghitung nilai prioritas kriteria (*priority vector*) dengan rumus jumlah kolom dibagi dengan jumlah kriteria. Perhitungannya dapat kita lihat pada Tabel 3.28.

Tabel 3. 28 Nilai prioritas kriteria (Priority vector)

Criteria	Priority vector
Late Frequency (%)	0.140
Bad Debts (IDR)	0.421
Paid Before Maturity (%)	0.047

Criteria	Priority vector
Total Payment (IDR)	0.131
Payment Complete Avg (month)	0.043
Order Frequency	0.217

Selanjutnya mencari nilai CR (*Consistensy Ratio*) untuk apakah penilaian perbandingan kriteria bersifat konsisten atau tidak. Ada 3 tahapan yang harus dilakukan, yaitu :

1. Menentukan nilai eigen maksimum (λ_{maks})

λ maks diperoleh dengan menjumlahkan hasil perkalian jumlah matriks kolom dengan *priority vector* untuk setiap kriteria. untuk jumlah matriks kolom dapat kita lihat pada tabel 3.27, untuk *priority vector* dapat kita lihat pada tabel 3.28, didapat nilai $\lambda_{maks} = 6.108$.

2. Menghitung Indeks Konsistensi (CI)

CI diperoleh dengan menghitung λ_{maks} dikurangi jumlah kriteria, kemudian dibagi dengan jumlah kriteria dikurangi 1. Sehingga didapat nilai CI = 0.02

3. Menghitung rasio konsistensi (CR)

CR diperoleh dengan membagi nilai CI dengan nilai RI. RI didapat dari daftar Indeks random konsistensi pada tabel 3.10. Untuk $n = 6$ adalah 1,24 dimana n adalah jumlah kriteria, maka didapat nilai CR = 0.071

Dari hasil perhitungan dapat diketahui bahwa nilai CR = 0.071, karena $CR < 0,1$, maka preferensi pembobotan adalah konsisten. Apabila nilai $CR > 0,1$ maka preferensi pembobotan tidak konsisten dan harus dilakukan pembobotan kembali hingga nilai $CR < 0,1$.

Setelah mendapatkan konsistensi dari hasil kriteria, selanjutnya menyusun alternatif-alternatif yang telah ditentukan

dalam bentuk matriks berpasangan untuk masing-masing kriteria. sehingga akan ada sebanyak n buah matriks berpasangan alternatif dimana nilai n adalah jumlah kriteria yang ada. Berikut data transaksi pelanggan yang menjadi alternatif dalam melakukan perankingan. Data tersebut ditunjukkan pada Tabel 3.29.

Tabel 3. 29. Data Pelanggan

alternatif / kriteria	Late Frequency (%)	Bad Debts (IDR)	Paid Before Maturity (%)	Total Payment (IDR)	Payment Avg (month)	Order Frequency
Customer 1	20	10000	10	200000	2	10
Customer 2	10	700	20	10000	1	5
Customer 3	1	1	66	300000	1	3
Customer 4	33	170	1	3000	1	3

Setelah diketahui nilai-nilai kriteria untuk setiap alternatif, selanjutnya dibandingkan setiap alternatif untuk setiap kriteria. Berikut penjelasan untuk setiap kriteria yang ada :

1. Late Frequency (%)

Pada kriteria presentase frekuensi keterlambatan pembayaran, nilai matriks diambil yang terkecil, sehingga untuk perhitungannya yaitu kolom dibagi baris. Apabila pelanggan belum pernah terlambat dalam melakukan pembayaran, maka nilai late frequency tersebut didefinisikan sebesar 0.001. Matriks perbandingan bebasangan terhadap kriteria pertama ditunjukkan pada Tabel 3.30 dan hasil normalisasinya ditunjukkan pada Tabel 3.31.

Tabel 3. 30. Pairwise comparison kriteria 1

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	0.5	0.05	1.65
Customer 2	2	1	0.1	3.3
Customer 3	20	10	1	33
Customer 4	0.606	0.303	0.0303	1

Tabel 3. 31. Normalisasi kriteria 1

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	0.5	0.05	1.65
Customer 2	2	1	0.1	3.3
Customer 3	20	10	1	33
Customer 4	0.606	0.303	0.0303	1

2. *Bad Debts (IDR)*

Pada kriteria jumlah piutang tak tertagih , nilai matriks perlu dibalik karena perusahaan menginginkan pelanggan dengan jumlah piutang yang terkecil. Apabila pelanggan tidak mempunyai piutang tak tertagih, maka nilai *bad debts* tersebut akan menjadi default 0.001 Matriks perbandingan bebasangan terhadap kriteria pertama ditunjukkan pada tabel 3.32 dan hasil normalisasinya ditunjukkan pada Tabel 3.33.

Tabel 3. 32. Pairwise comparison kriteria 2

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	0.5	0.05	1.65
Customer 2	2	1	0.1	3.3
Customer 3	20	10	1	33
Customer 4	0.606	0.303	0.030	1

Tabel 3. 33. Normalisasi kriteria 2

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	0.042	0.042	0.042	0.042
Customer 2	0.085	0.085	0.085	0.085
Customer 3	0.847	0.847	0.847	0.847
Customer 4	0.026	0.026	0.026	0.026

3. *Paid Before Maturity (%)*

Untuk kriteria presentase pelanggan melunasi hutang sebelum tanggal jatuh tempo, perusahaan menginginkan pelunasan paling cepat. Apabila pelanggan tidak pernah melunasi hutang sebelum jatuh tempo, maka nilai kriteria *paid before maturity* akan menjadi default 0.001. Dapat

dilihat pairwise comparison pada Tabel 3.34 dan hasil normalisasi pada Tabel 3.35

Tabel 3. 34. Pairwise comparison kriteria 3

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	0.5	0.152	10
Customer 2	2	1	0.303	20
Customer 3	6.6	3.3	1	66
Customer 4	0.1	0.05	0.015	1

Tabel 3. 35. Normalisasi kriteria 3

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	0.103	0.103	0.103	0.103
Customer 2	0.206	0.206	0.206	0.206
Customer 3	0.680	0.680	0.680	0.680
Customer 4	0.010	0.010	0.010	0.010

4. Total Order (IDR)

Pada kriteria *total order*, perusahaan mengingingkan pelanggan dengan *total order* terbanyak. Dapat dilihat hasil matriks perbandingan berpasangan pada Tabel 3.36 dan hasil normalisasi pada Tabel 3.37.

Tabel 3. 36. Pairwise comparison kriteria 4

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	20	0.667	66.667
Customer 2	0.05	1	0.0333	3.333
Customer 3	1.5	30	1	100
Customer 4	0.015	0.3	0.01	1

Tabel 3. 37. Normalisasi kriteria 4

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	0.389	0.389	0.389	0.389
Customer 2	0.019	0.019	0.019	0.019
Customer 3	0.585	0.585	0.585	0.585
Customer 4	0.005	0.005	0.005	0.005

5. *Payment Average (month)*

Pada kriteria rata-rata waktu transaksi sampai berstatus lunas, perusahaan menginginkan pelanggan dengan rata-rata pembayaran paling cepat. Dapat dilihat hasil perbandingan matriks berpasangan pada Tabel 3.38 dan hasil normalisasi pada Tabel 3.39.

Tabel 3. 38. *Pairwise comparison* kriteria 5

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	0.5	0.5	0.5
Customer 2	2	1	1	1
Customer 3	2	1	1	1
Customer 4	2	1	1	1

Tabel 3. 39. Normalisasi kriteria 5

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	0.143	0.143	0.143	0.143
Customer 2	0.286	0.286	0.286	0.286
Customer 3	0.286	0.286	0.286	0.286
Customer 4	0.286	0.286	0.286	0.286

7. *Order frequency*

Pada kriteria *order frequency*, perusahaan menginginkan pelanggan dengan *frekuensi order* paling banyak dan berstatus lunas. Dapat dilihat hasil matriks perbandingan pada Tabel 3.40. dan hasil normalisasi pada Tabel 3.41

Tabel 3. 40. *Pairwise comparison* kriteria 6

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	1	2	3.333	3.333
Customer 2	0.5	1	1.667	1.667
Customer 3	0.3	0.6	1	1
Customer 4	0.3	0.6	1	1

Tabel 3. 41. Normalisasi kriteria 6

	Customer 1	Customer 2	Customer 3	Customer 4
Customer 1	0.476	0.476	0.476	0.476
Customer 2	0.238	0.238	0.238	0.238
Customer 3	0.143	0.143	0.143	0.143
Customer 4	0.143	0.143	0.1423	0.143

Setelah mendapatkan *priority vector* setiap kriteria untuk ketiga alternatif tersebut, Langkah terakhir adalah menghitung prioritas global untuk ketiga alternatif tersebut sebagai nilai yang digunakan oleh pengambil keputusan berdasarkan skor yang tertinggi karena merupakan pelanggan dengan tingkat loyalitas paling tinggi. Total skor dari masing-masing pelanggan ditunjukkan pada Tabel 3.42.

Tabel 3. 42. Total Skor Pelanggan

	Customer 1	Customer 2	Customer 3	Customer 4
Late Frequency (%)	0.042	0.085	0.026	0.042
Bad Debts	0.059	0.001	0.006	0.059
Paid Before Maturity (%)	0.103	0.206	0.010	0.103
Total Payment	0.389	0.019	0.006	0.389
Total Payment	0.143	0.286	0.286	0.143
Payment Complete Average(month)	0.476	0.238	0.143	0.476
Skor	0.171	0.088	0.689	0.051

Rumus prioritas global/skor adalah dengan menjumlahkan perkalian dari *priority vector* setiap kriteria pada alternatif dengan jumlah baris kriteria yang sudah dihitung sebelumnya. Dapat kita lihat pada tabel 3.42 rincian *priority vector* setiap alternatif dan kriteria, serta jumlah baris untuk tiap kriteria dan hasil prioritas global untuk ketiga alternatif. Rincian perhitungan untuk prioritas global dapat kita lihat dibawah ini :

- Prioritas global Customer 1 = 0.171
- Prioritas global Customer 2 = 0.088

- Prioritas global Customer 3 = 0.689
- Prioritas global Customer 4 = 0.051

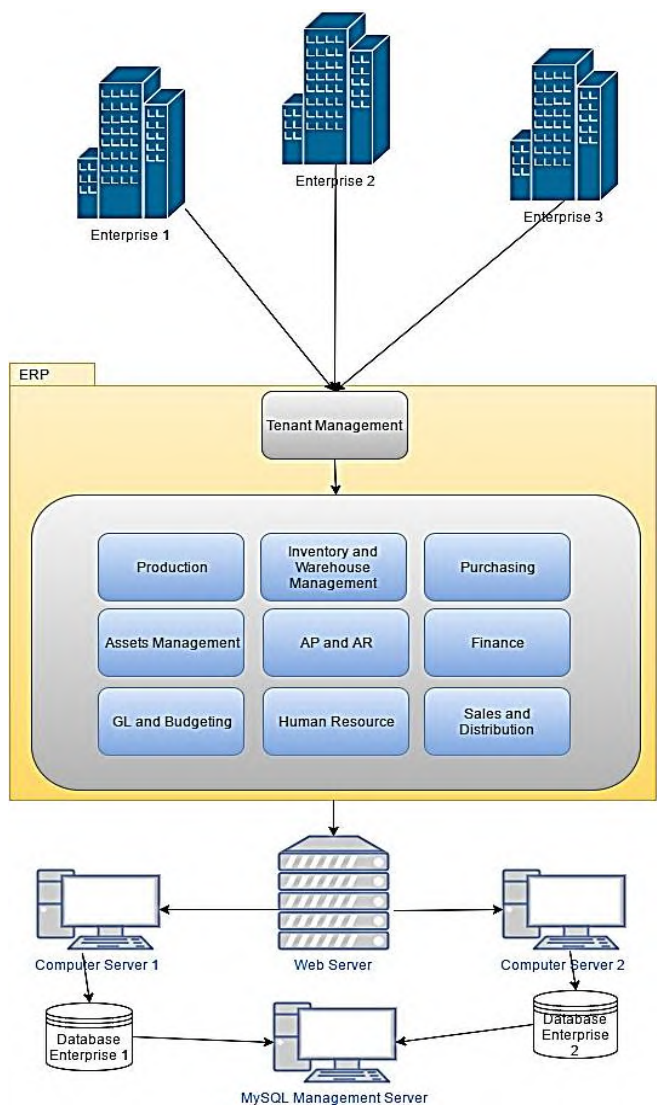
Dari hasil keseluruhan perhitungan, dapat kita simpulkan bahwa Customer 3 mempunyai nilai prioritas global yang paling tinggi, yaitu 0.689 dan berada pada urutan pertama sebagai prioritas Customer terbaik. Customer 1 dengan nilai prioritas global sebesar 0.171. Customer 2 dengan nilai prioritas global sebesar 0.088 dan yang terakhir adalah Customer 4 dengan nilai prioritas global sebesar 0.051.

3.2. Deskripsi Umum Sistem

Pada Tugas Akhir ini dibangun aplikasi *Account Payable*, *Account Receivable* yang terintegrasi ke dalam sebuah sistem Enterprise Resource Planning (ERP). Tujuan dari aplikasi *Account Payable* dan *Account Receivable* adalah untuk pengelolaan data keuangan perusahaan, khususnya di bagian hutang piutang. Kedua modul fungsi tersebut terintegrasikan dengan modul-modul fungsi yang lain dalam sebuah sistem ERP. Sistem yang dibangun berorientasi *multitenancy* dengan basis data terdistribusi. Desain basis data terdistribusi dan *multitenancy* yang digunakan dalam sistem ini ditunjukkan pada Gambar 3.9.

Pada Gambar 3.9, dijelaskan bahwa perusahaan 1, perusahaan 2, dan perusahaan 3 dapat menggunakan satu aplikasi ERP secara bersamaan. Setiap perusahaan mempunyai 2 basis data yang telah direplikasi yaitu *Computer Server 1* dengan IP *Enterprise 1* dan *Computer Server 2* dengan IP *Enterprise 2*. Komputer 3 berfungsi sebagai *server cluster* untuk melakukan replikasi, sehingga apabila basis data 1 atau *Computer Server 1* dalam keadaan mati, sistem ERP masih tetap berjalan.

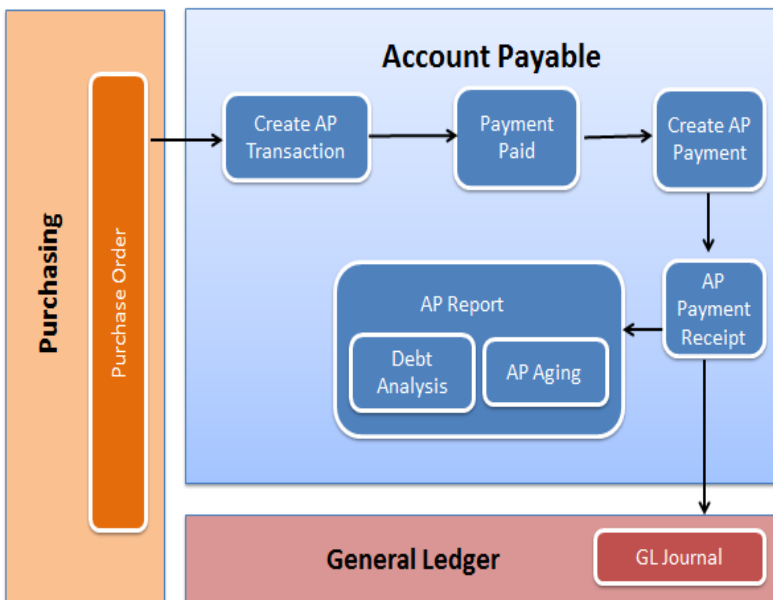
Berikut dijelaskan deskripsi umum sistem yang terdapat pada modul *Account Payable* dan *Account Receivable* yang dijelaskan pada subbab 3.2.1. dan 3.2.2.



Gambar 3. 10. Arsitektur Basis Data Terdistribusi

3.2.1. *Account Payable*

Account Payable bertanggung jawab dalam mencatat jumlah hutang yang harus dibayarkan oleh perusahaan kepada pemasok atau *supplier*. Apabila perusahaan telah menerima tagihan pembayaran dari *supplier*, maka tagihan tersebut akan ditambahkan ke dalam akun hutang dagang atau *account payable*, dan kemudian akan dicatat pembayarannya ketika hutang tersebut telah dibayarkan. Pada ERP 2016, pada modul *account payable* ditambahkan sistem perangkingan hutang menggunakan metode AHP (*Analytical Hierarchy Process*) yang telah dijabarkan sebelumnya pada sub-bab 3.1.3 Proses bisnis *account payable* ini dibagi menjadi dua subproses bisnis, yaitu *Account Payable Transaction* dan *Account Payable Payment*. Berikut ditampilkan alur kerja modul *Account Payable* pada Gambar 3.10.



Gambar 3. 11. Alur Kerja *Account Payable*

3.2.1.1. *Account Payable Transaction*

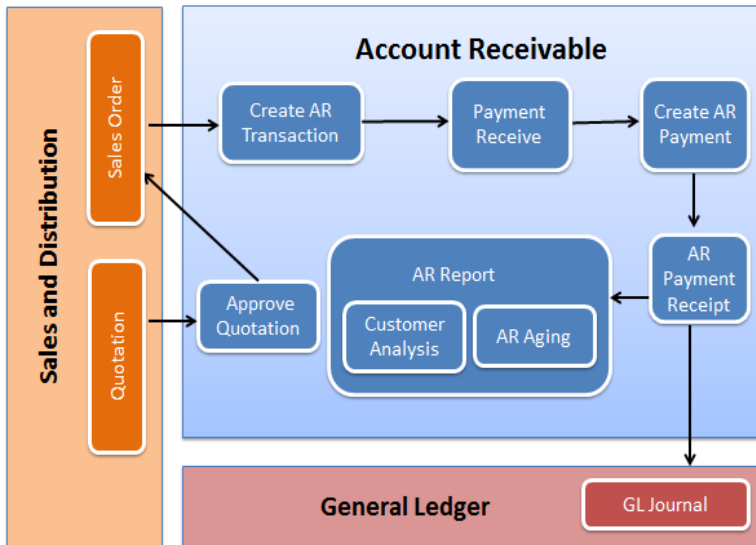
Subproses bisnis *Account Payable Transaction* bertugas mencatat jumlah tagihan yang harus dibayar perusahaan kepada *supplier* setelah dilakukan *purchase order* atau penerimaan barang oleh bagian *Purchasing*. Data yang dimasukkan pada *Account Payable Transaction* mengacu pada data yang ada pada *Purchase Order*. Tagihan tersebut kemudian digunakan untuk menerbitkan *Account Payable Payment* yang dilakukan untuk pembayaran hutang perusahaan.

3.2.1.2. *Account Payable Payment*

Subproses bisnis *Account Payable Payment* bertugas mencatat pembayaran yang dilakukan kepada *supplier*. Proses ini dilakukan apabila perusahaan melakukan pembayaran hutang kepada pemasok. Pembayaran hutang dapat dilakukan secara langsung/tunai dan dapat dilakukan secara bertahap/kredit. Setiap pembayaran yang dilakukan, baik lunas maupun kredit langsung tercatat otomatis ke dalam jurnal *General Ledger*.

3.2.1.3. *Account Receivable*

Account Receivable mencatat jumlah uang atau tagihan yang harus dibayarkan oleh pelanggan kepada perusahaan karena membeli barang atau jasa secara kredit yang nantinya akan menjadi pemasukan bagi keuangan perusahaan. Proses bisnis ini dibagi menjadi dua subproses bisnis, yaitu *Account Receivable Transaction* dan *Account Receivable Payment*. Pada ERP 2016, pada modul *account receivable* ditambahkan sistem perangkingan pelanggan. Sistem perangkingan tersebut menggunakan metode AHP (*Analytical Hierarchy Process*) yang telah dijabarkan pada sub-bab 3.1.4. Berikut ditampilkan alur kerja modul *Account Receivable* pada Gambar 3.11.



Gambar 3. 12. Alur kerja Account Receivable

3.2.1.3.1. Account Receivable Transaction

Subproses bisnis *Account Receivable Transaction* bertugas mencatat jumlah tagihan yang harus dibayar perusahaan kepada supplier setelah dilakukan *sales order* atau penjualan barang oleh bagian *Sales*. Tagihan tersebut kemudian digunakan untuk menerbitkan *Account Payable Payment* yang dilakukan untuk mencatat pembayaran dari pelanggan.

3.2.1.3.2. Account Receivable Payment

Subproses bisnis *Sales Account Receivable Payment* bertugas untuk mencatat pembayaran yang dilakukan oleh pelanggan. Proses ini dilakukan apabila pelanggan melakukan pembayaran piutang kepada perusahaan. Setiap pembayaran yang dilakukan, baik lunas maupun kredit langsung dicatat ke dalam jurnal *General Ledger*. Melalui proses ini, piutang perusahaan akan berkurang.

3.3. Identifikasi Pengguna

Berdasarkan deskripsi umum yang telah dijelaskan pada Subbab 3.2, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini dua, yaitu staf keuangan dan manajer keuangan. Penjelasan mengenai pengguna yang juga disebut aktor dalam sistem, dijelaskan pada Tabel 3.43.

Tabel 3. 43. Pengguna Aplikasi

No.	Aktor	Deskripsi
1.	<i>Staf Keuangan</i>	Orang yang bertanggung jawab atas proses-proses pengelolaan data pada subdomain <i>Account Payable</i> dan <i>Account Receivable</i> .
2.	<i>Manajer Keuangan</i>	Orang yang bertanggung jawab atas persetujuan terhadap proses-proses pada subdomain <i>Account Payable</i> dan <i>Account Receivable</i> .

3.4. Spesifikasi Kebutuhan Perangkat Lunak

Berdasarkan uraian mengenai cakupan perangkat lunak pada subbab 3.1.1, maka terdapat beberapa spesifikasi kebutuhan perangkat lunak yang harus dipenuhi agar sistem mampu mengakomodasi permasalahan yang diberikan dan bisa bekerja dengan baik. Spesifikasi kebutuhan perangkat lunak pada Tugas Akhir ini terdiri atas kebutuhan fungsional.

3.4.1. Kebutuhan Fungsional

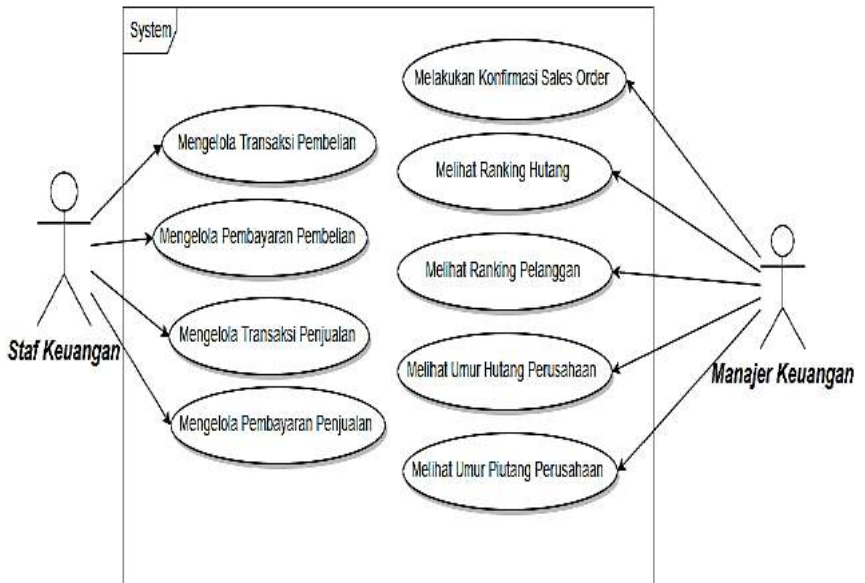
Kebutuhan fungsional berisi kebutuhan utama yang harus dipenuhi oleh sistem agar sistem mampu bekerja dengan baik. Kebutuhan fungsionalitas juga secara tidak langsung mendeskripsikan fungsionalitas perangkat lunak. Daftar kebutuhan fungsionalitas dapat dilihat pada Tabel 3.44.

Tabel 3. 44 Daftar kebutuhan fungsional perangkat lunak

No.	Kebutuhan Fungsional	Deskripsi
F-001	Mencatat tagihan pembelian	Pengguna dapat membuat dan mencatat data tagihan pembelian barang dari pemasok
F-002	Mencatat pembayaran pembelian	Pengguna dapat membuat dan mencatat data pembayaran hutang yang telah dibayarkan kepada pemasok.
F-003	Mencatat tagihan penjualan	Pengguna dapat membuat dan mencatat data tagihan penjualan barang kepada pelanggan
F-004	Mencatat pembayaran dari penjualan	Pengguna dapat membuat dan mencatat data pembayaran piutang yang dibayarkan oleh pelanggan.
F-005	Melakukan konfirmasi terhadap <i>quotation/sales order</i>	Pengguna dapat menyetujui atau menolak <i>quotation/sales order</i>

3.5. Kasus Penggunaan

Pada bagian ini akan dibahas secara rinci untuk masing-masing kasus penggunaan. Pada Tugas Akhir ini, terdapat kasus penggunaan yang ditunjukkan pada Gambar 3.12 dan Tabel 3.45. Berdasarkan kebutuhan fungsional F-001 dan F-002, ditambahkan fitur tambahan yaitu melihat laporan umur hutang dan melihat laporan ranking hutang yang dijelaskan pada Kasus Penggunaan UC-006 dan UC-007. Pada kebutuhan fungsional F-003 dan F-004, maka ditambahkan fitur tambahan yaitu melihat laporan umur piutang dan melihat laporan ranking pelanggan yang dijelaskan pada Kasus Penggunaan UC-008 dan UC-009. Sedangkan pada kebutuhan fungsional F-005 telah terakomodasi oleh Kasus Penggunaan UC-005 yaitu melakukan persetujuan terhadap *sales order*.



Gambar 3. 13. Diagram kasus penggunaan aplikasi

Tabel 3. 45. Deskripsi Kasus Penggunaan Aplikasi

No.	Kode	Nama Kasus Penggunaan	Deskripsi
1.	UC-001	Mengelola transaksi pembelian	Melihat, menambah, menyunting, serta menghapus data transaksi pembelian.
2.	UC-002	Mengelola pembayaran pembelian	Melihat, menambah, menyunting, serta menghapus data pembayaran pembelian.
3.	UC-003	Mengelola transaksi penjualan	Melihat, menambah, menyunting, serta menghapus data transaksi penjualan
4.	UC-004	Mengelola pembayaran penjualan barang	Melihat, menambah, menyunting, serta menghapus data pembayaran penjualan barang.

No.	Kode	Nama Kasus Penggunaan	Deskripsi
5.	UC-005	Melakukan persetujuan terhadap sales order	Menyetujui atau menolak sales order
6.	UC-006	Melihat laporan umur hutang	Melihat laporan umur hutang perusahaan
7.	UC-007	Melihat laporan umur piutang	Melihat laporan umur piutang perusahaan
8.	UC-008	Melihat laporan ranking hutang	Melihat laporan ranking hutang perusahaan
9.	UC-009	Melihat laporan ranking pelanggan	Melihat laporan ranking piutang pelanggan.

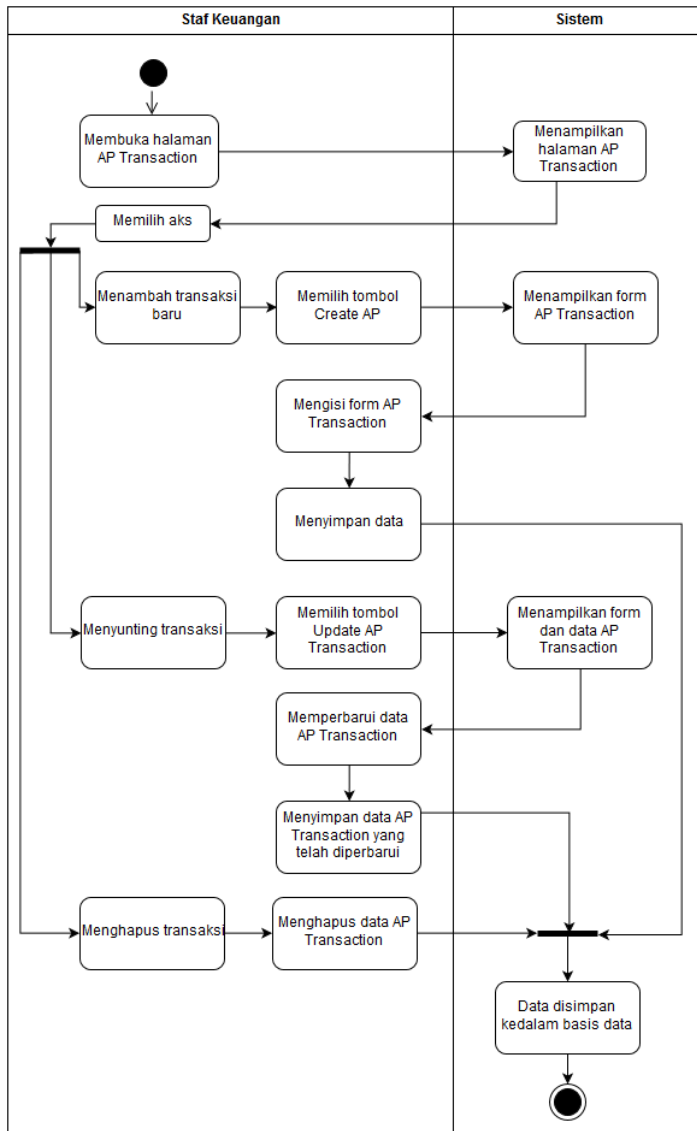
3.5.1. Kasus Penggunaan Mengelola Transaksi Pembelian (UC-001)

Pada kasus penggunaan ini, pengguna dapat melihat, menambah, menyunting, dan menghapus data transaksi pembelian. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.46, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.13.

Tabel 3. 46. Rincian kasus penggunaan mengelola transaksi pembelian

Nama	Mengelola transaksi pembelian
Nomor	UC-001
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data transaksi pembelian
Tipe	Fungsional
Aktor	Staf Keuangan

Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar transaksi pembelian
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar transaksi pembelian 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat transaksi pembelian baru A2. Pengguna memilih menyunting transaksi tertentu A3. Pengguna memilih menghapus transaksi tertentu 3. Kasus penggunaan berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih membuat transaksi pembelian baru <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan transaksi pembelian 2. Pengguna memasukkan data pembuatan transaksi pembelian baru 3. Sistem menyimpan transaksi pembelian baru 4. Berlanjut ke alur normal langkah 3 A2. Pengguna memilih menyunting transaksi pembelian tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting transaksi pembelian 2. Pengguna menyunting data transaksi pembelian 3. Sistem memperbaharui data transaksi pembelian 4. Berlanjut ke alur normal langkah 3 A3. Pengguna memilih menghapus transaksi pembelian tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan transaksi pembelian 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data transaksi pembelian 4. Berlanjut ke alur normal langkah 3



Gambar 3. 14. Diagram aktivitas mengelola transaksi pembelian

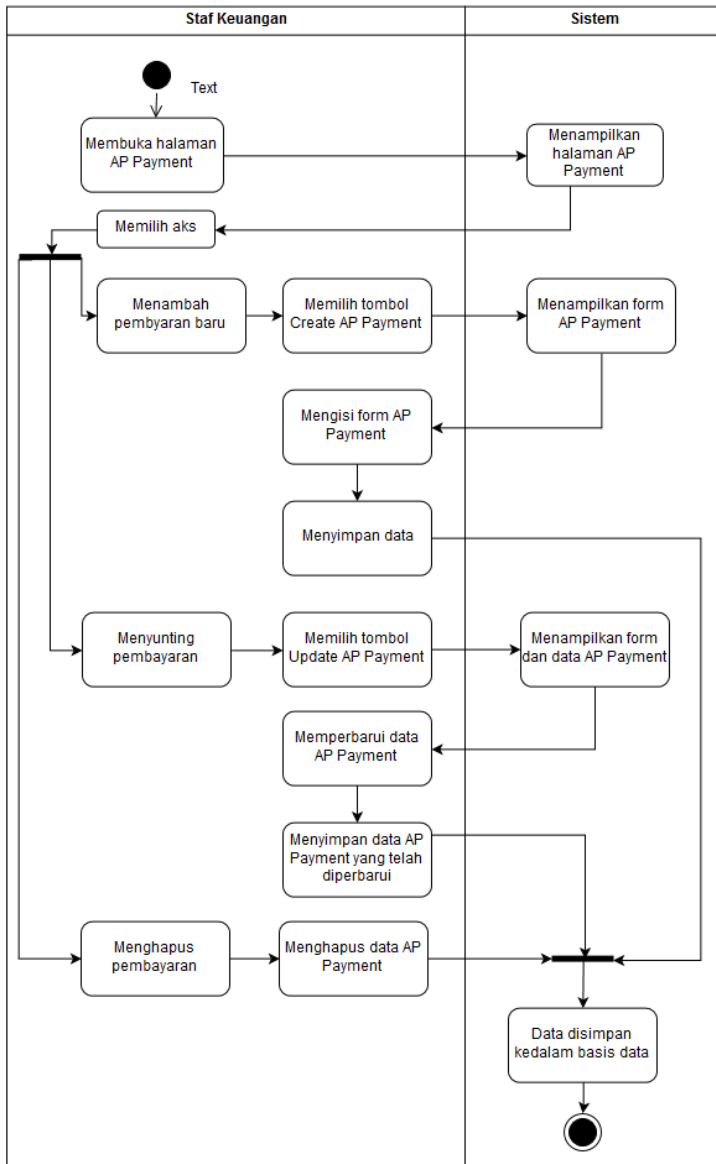
3.5.2. Kasus Penggunaan Mengelola Pembayaran Pembelian (UC-002)

Pada kasus penggunaan ini, pengguna dapat melihat, menambah, menyunting, dan menghapus data pembayaran pembelian. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.47, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.14.

Tabel 3. 47. Rincian kasus penggunaan mengelola pembayaran pembelian

Nama	Mengelola pembayaran pembelian
Nomor	UC-002
Deskripsi	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data pembayaran pembelian
Tipe	Fungsional
Aktor	Staf Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar pembayaran pembelian
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pembayaran pembelian 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat pembayaran pembelian baru A2. Pengguna memilih menyunting pembayaran tertentu A3. Pengguna memilih menghapus pembayaran tertentu 3. Kasus penggunaan berakhir

Alur Alternatif	<p>A1. Pengguna memilih membuat pembayaran pembelian baru</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan pembayaran pembelian 2. Pengguna memasukkan data pembuatan pembayaran pembelian baru 3. Sistem menyimpan pembayaran pembelian baru 4. Berlanjut ke alur normal langkah 3 <p>A2. Pengguna memilih menyunting pembayaran pembelian tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting pembayaran pembelian 2. Pengguna menyunting data pembayaran pembelian 3. Sistem memperbaharui data pembayaran pembelian 4. Berlanjut ke alur normal langkah 3 <p>A3. Pengguna memilih menghapus pembayaran pembelian tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan pembayaran pembelian 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data pembayaran pembelian 4. Berlanjut ke alur normal langkah 3
------------------------	---



Gambar 3. 15. Diagram aktivitas mengelola pembayaran pembelian

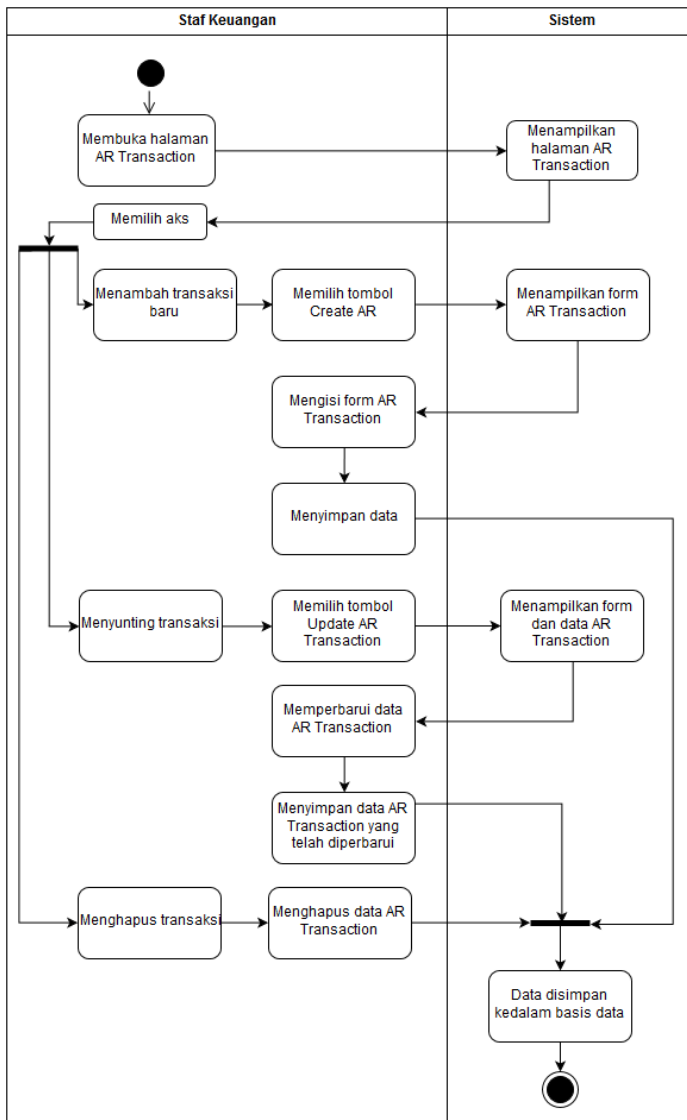
3.5.3. Kasus Penggunaan Mengelola Transaksi Penjualan (UC-003)

Pada kasus penggunaan ini, pengguna dapat melihat, menambah, menyunting, dan menghapus data transaksi penjualan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.48, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.15.

Tabel 3. 48. Rincian kasus penggunaan mengelola transaksi penjualan

Nama	Mengelola transaksi penjualan
Nomor	UC-003
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data transaksi penjualan
Tipe	Fungsional
Aktor	Staf Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar transaksi penjualan
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar transaksi penjualan 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat transaksi penjualan baru A2. Pengguna memilih menyunting transaksi tertentu A3. Pengguna memilih menghapus transaksi tertentu 3. Kasus penggunaan berakhir

Alur Alternatif	<p>A1. Pengguna memilih membuat transaksi penjualan baru</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan transaksi penjualan 2. Pengguna memasukkan data pembuatan transaksi penjualan baru 3. Sistem menyimpan transaksi penjualan baru 4. Berlanjut ke alur normal langkah 3 <p>A2. Pengguna memilih menyunting transaksi penjualan tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting transaksi penjualan 2. Pengguna menyunting data transaksi penjualan 3. Sistem memperbaharui data transaksi penjualan 4. Berlanjut ke alur normal langkah 3 <p>A3. Pengguna memilih menghapus transaksi penjualan tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan transaksi penjualan 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data transaksi penjualan 4. Berlanjut ke alur normal langkah 3
------------------------	--



Gambar 3. 16. Diagram aktivitas mengelola transaksi penjualan

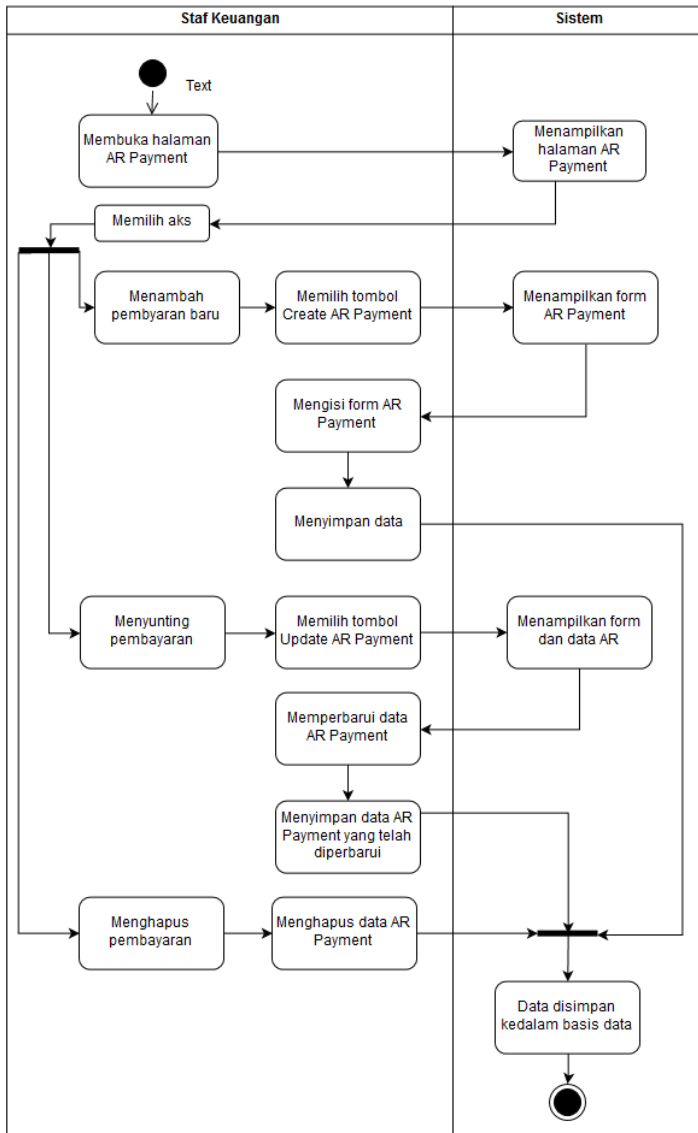
3.5.4. Kasus Penggunaan Mengelola Pembayaran Penjualan (UC-004)

Pada kasus penggunaan ini, pengguna dapat melihat, menambah, menyunting, dan menghapus data pembayaran penjualan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.49, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.16.

Tabel 3. 49. Rincian kasus penggunaan mengelola pembayaran penjualan

Nama	Mengelola pembayaran penjualan
Nomor	UC-004
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data pembayaran penjualan
Tipe	Fungsional
Aktor	Staf Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar pembayaran penjualan
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pembayaran penjualan 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat pembayaran penjualan baru A2. Pengguna memilih menyunting pembayaran tertentu A3. Pengguna memilih menghapus pembayaran tertentu 3. Kasus penggunaan berakhir

Alur Alternatif	<p>A1. Pengguna memilih membuat pembayaran penjualan baru</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan pembayaran penjualan 2. Pengguna memasukkan data pembuatan pembayaran penjualan baru 3. Sistem menyimpan pembayaran penjualan baru 4. Berlanjut ke alur normal langkah 3 <p>A2. Pengguna memilih menyunting pembayaran penjualan tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting pembayaran pembelian 2. Pengguna menyunting data pembayaran penjualan 3. Sistem memperbaharui data pembayaran penjualan 4. Berlanjut ke alur normal langkah 3 <p>A3. Pengguna memilih menghapus pembayaran penjualan tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan pembayaran penjualan 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data pembayaran penjualan 4. Berlanjut ke alur normal langkah 3
------------------------	---



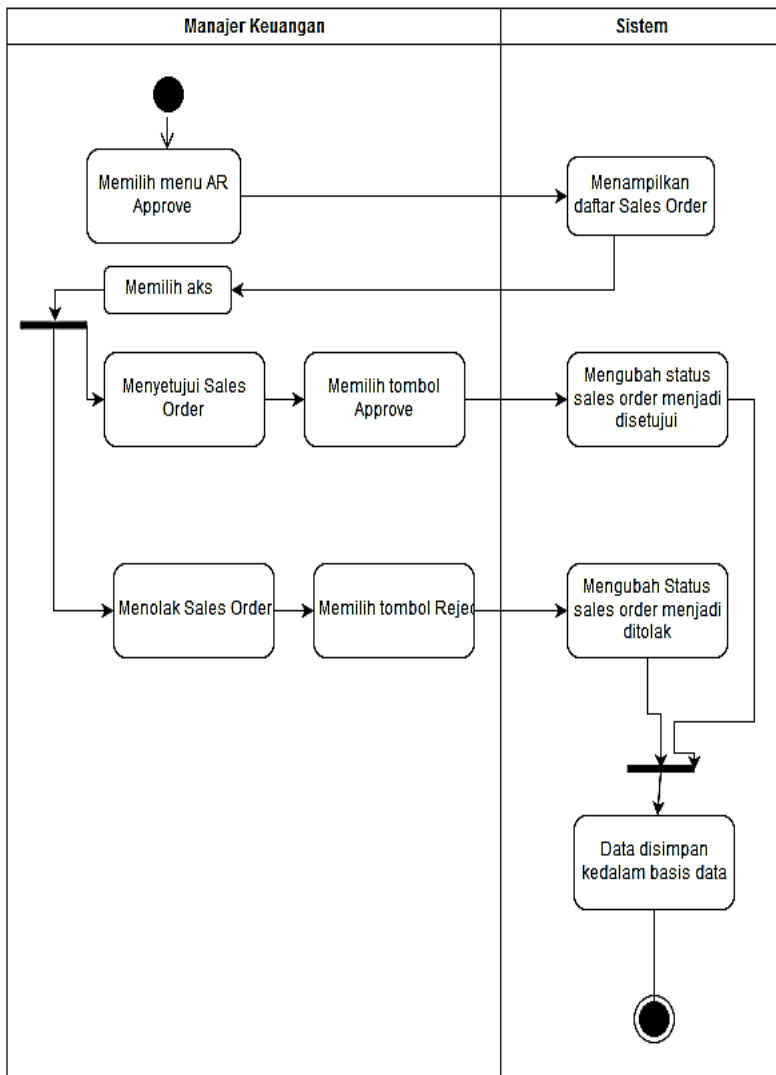
Gambar 3. 17. Diagram aktivitas mengelola pembayaran penjualan

3.5.5. Kasus Penggunaan Melakukan Konfirmasi *Sales Order* (UC-005)

Pada kasus penggunaan ini, pengguna dapat menyetujui *sales order*. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.50, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.17.

Tabel 3. 50. Rincian kasus penggunaan melakukan konfirmasi *Sales Order*

Nama	Melakukan konfirmasi <i>Sales Order</i>
Nomor	UC-005
Description	Kasus penggunaan ini digunakan untuk melakukan konfirmasi <i>sales order</i>
Tipe	Fungsional
Aktor	Manajer Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem mengubah status <i>sales order</i>
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>sales order</i> yang belum disetujui 2. Pengguna memilih kegiatan yang dilakukan <ol style="list-style-type: none"> A1. Pengguna menyetujui <i>sales order</i> tertentu A2. Pengguna menolak <i>sales order</i> tertentu 3. Kasus penggunaan berakhir
Alur Alternatif	<ol style="list-style-type: none"> A.2. Pengguna menyetujui <i>sales order</i> tertentu <ol style="list-style-type: none"> 1. Sistem mengubah status <i>sales order</i> menjadi disetujui 2. Kembali ke alur normal langkah 3 A.2. Pengguna menolak <i>sales order</i> tertentu <ol style="list-style-type: none"> 1. Sistem mengubah status <i>sales order</i> menjadi ditolak 2. Kembali ke alur normal langkah 3



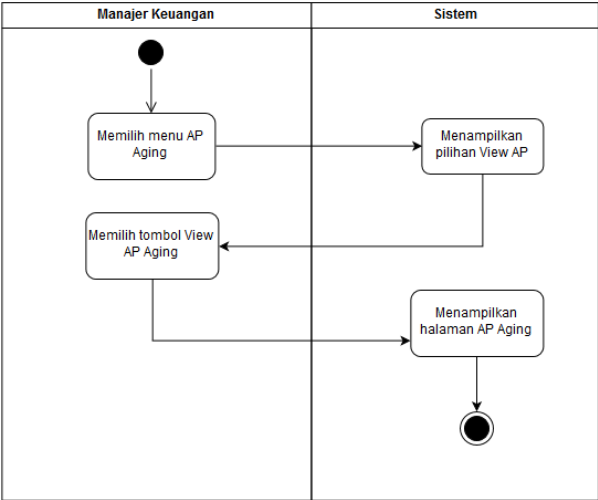
Gambar 3. 18. Diagram aktivitas melakukan konfirmasi sales order

3.5.6. Kasus Penggunaan Melihat Umur Hutang Perusahaan (UC-006)

Pada kasus penggunaan ini, pengguna dapat melihat data hutang perusahaan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.51, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.18.

Tabel 3. 51. Rincian kasus penggunaan melihat umur hutang perusahaan

Nama	Melihat umur hutang perusahaan
Nomor	UC-006
Deskripsi	Kasus penggunaan ini digunakan untuk melihat umur hutang perusahaan pada setiap pemasok.
Tipe	Fungsional
Aktor	Manajer Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar umur hutang perusahaan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan lihat umur hutang 2. Sistem menampilkan umur hutang perusahaan 3. Kasus penggunaan berakhir
Alur Alternatif	-



Gambar 3. 19. Diagram aktivitas melihat umur hutang perusahaan

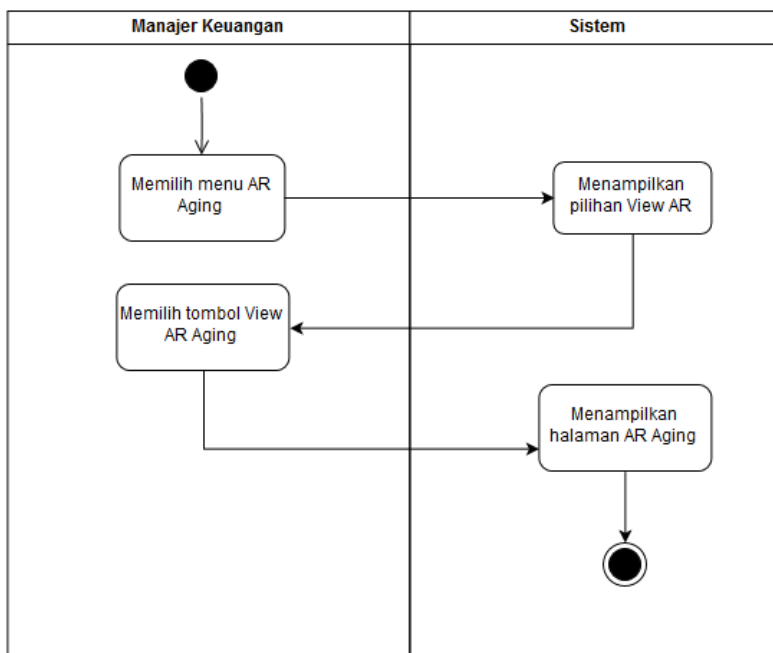
3.5.7. Kasus Penggunaan Melihat Umur Piutang Perusahaan (UC-007)

Pada kasus penggunaan ini, pengguna dapat melihat data hutang perusahaan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.52, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.19.

Tabel 3. 52. Rincian kasus penggunaan melihat umur piutang perusahaan

Nama	Melihat umur piutang perusahaan
Nomor	UC-007
Deskripsi	Kasus penggunaan ini digunakan untuk melihat umur piutang perusahaan pada setiap pelanggan.
Tipe	Fungsional
Aktor	Manajer Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem

Kondisi Akhir	Sistem menampilkan daftar umur piutang perusahaan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan lihat umur piutang 2. Sistem menampilkan umur piutang perusahaan 3. Kasus penggunaan berakhir
Alur Alternatif	-



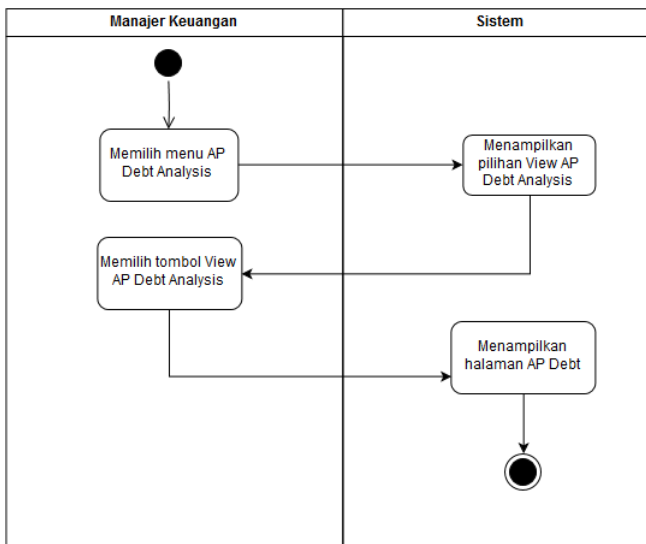
Gambar 3. 20. Diagram aktivitas melihat umur piutang perusahaan

3.5.8. Kasus Penggunaan Melihat Ranging Hutang Perusahaan (UC-008)

Pada kasus penggunaan ini, pengguna dapat melihat data hutang perusahaan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.53, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.20.

Tabel 3. 53. Rincian kasus penggunaan melihat ranking hutang perusahaan

Nama	Melihat ranking hutang perusahaan
Nomor	UC-008
Description	Kasus penggunaan ini digunakan untuk melihat ranking hutang perusahaan
Tipe	Fungsional
Aktor	Manajer Keuangan
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar ranking hutang perusahaan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan lihat ranking hutang 2. Sistem menampilkan ranking hutang perusahaan 3. Kasus penggunaan berakhir
Alur Alternatif	-



Gambar 3. 21. Diagram aktivitas melihat ranking hutang perusahaan

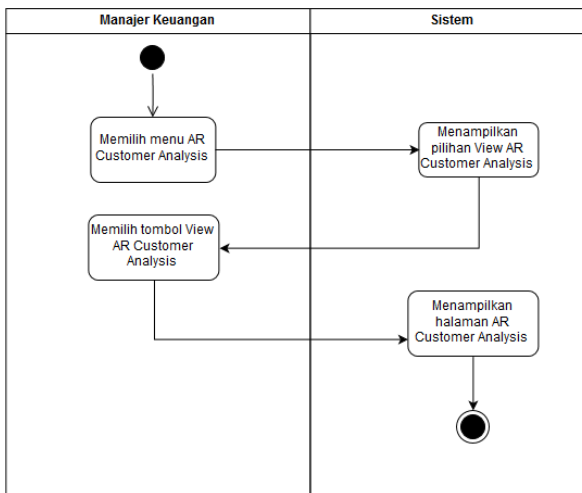
3.5.9. Kasus Penggunaan Melihat Rangking Pelanggan (UC-009)

Pada kasus penggunaan ini, pengguna dapat melihat data ranking pelanggan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.54, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.21.

Tabel 3. 54. Rincian kasus penggunaan melihat ranking pelanggan

Nama	Melihat ranking pelanggan
Nomor	UC-009
Deskripsi	Kasus penggunaan ini digunakan untuk melihat ranking pelanggan
Tipe	Fungsional
Aktor	Manajer Keuangan

Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan daftar ranking pelanggan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan lihat ranking pelanggan 2. Sistem menampilkan ranking pelanggan 3. Kasus penggunaan berakhir
Alur Alternatif	-



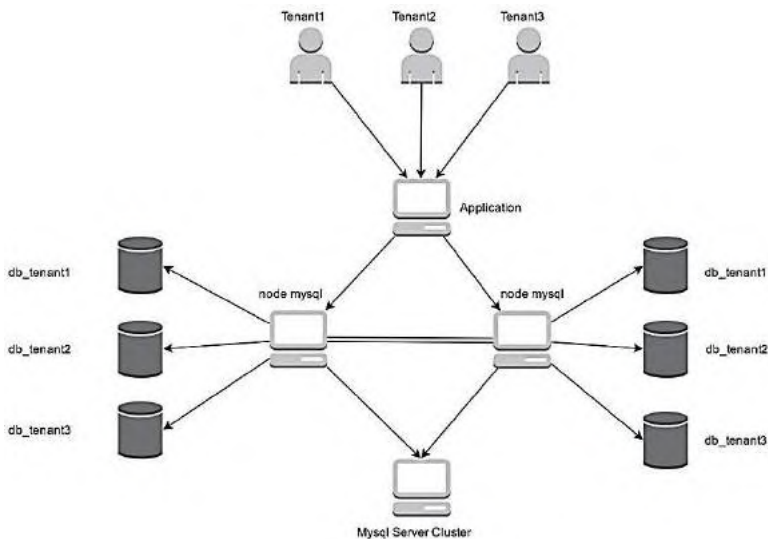
Gambar 3. 22. Diagram aktivitas melihat ranking pelanggan

3.6. Perancangan

Tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan *multitenancy*, perancangan basis data, perancangan RBAC, dan perancangan antarmuka sistem. Secara rinci mengenai perancangan perangkat lunak sebagai berikut:

3.6.1. Perancangan *Multitenancy*

Pada bagian ini akan dijelaskan mengenai perancangan multi-tenancy yang terdapat pada sistem ERP 2016. Secara rinci mengenai *multi-tenancy* dijabarkan pada Gambar 3.22.



Gambar 3. 23. Perancangan *Multi-tenancy*

Gambar 3.16. diatas menerangkan bagaimana alur multi-tenancy yang terjadi pada sistem ERP. Pembagian node mysql sudah dijelaskan pada sub-bab sebelumnya yaitu dengan cara *clustering* menjadi 2 database yang berbeda. Basis data terdistribusi merupakan penunjang dari *multitenancy*, yang berfungsi untuk mengatur *session* setiap perusahaan yang akan mengakses sistem ERP.

3.6.2. Perancangan Basis Data

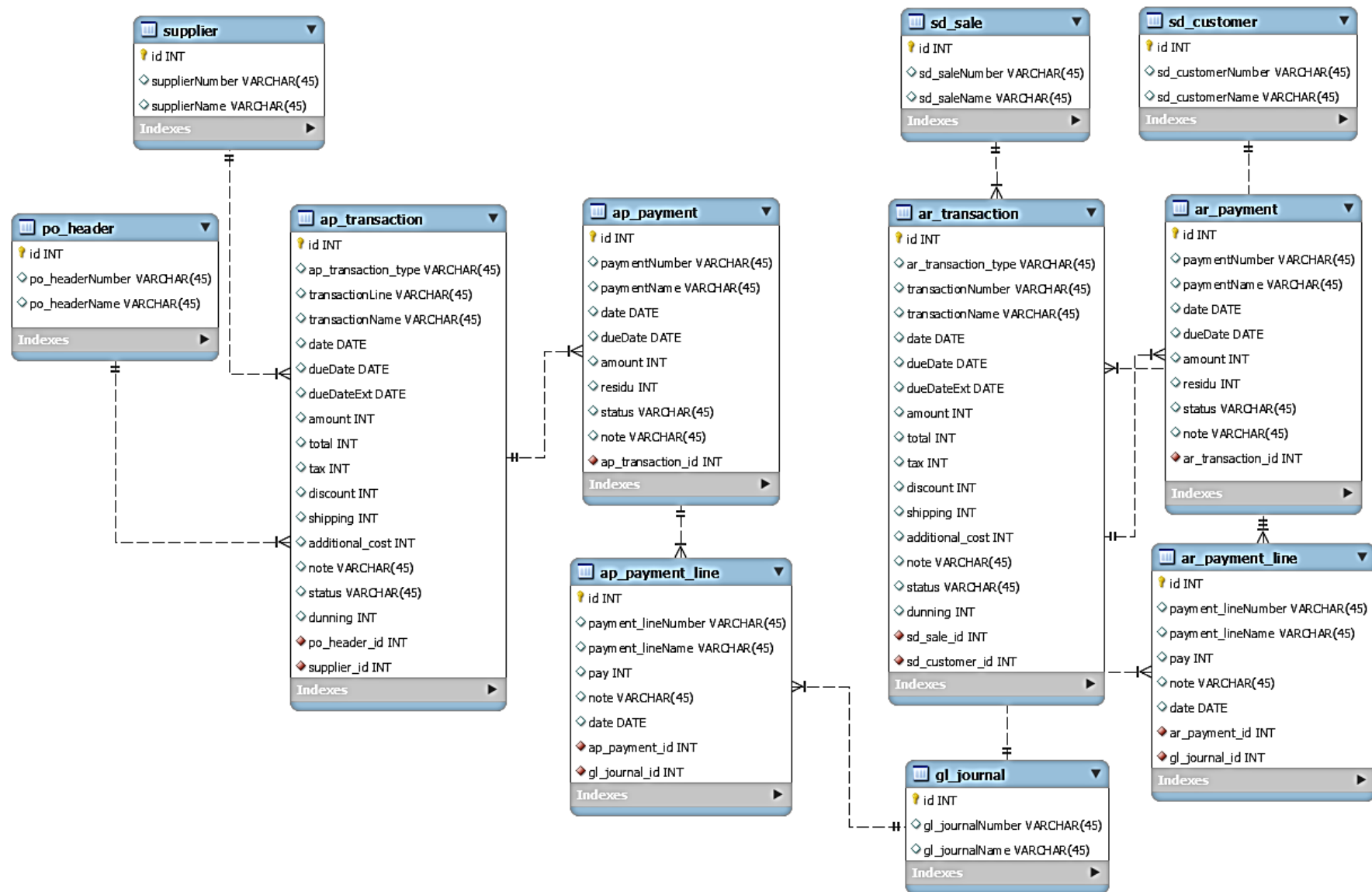
Pada Subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada modul *Account Payable* dan *Account Receivable*. Terdapat 6 tabel utama yang merupakan tabel transaksi yang terdapat pada modul *Account Payable* dan *Account Receivable*, antara lain Tabel **ap_transaction**, **ap_payment**, **ap_payment_line**, **ar_transaction**, **ar_payment**, **ar_payment_line**. Tabel-tabel transaksi tersebut berhubungan dengan beberapa tabel master dan tabel transaksi dari modul lain, diantaranya terdapat Tabel **supplier** yang merupakan tabel master, Tabel **po_header** yang merupakan tabel transaksi dari modul *Purchasing*, Tabel **sd_sale** dan **sd_customer** yang berasal dari modul *Sales and Distribution*, dan Tabel **gl_journal** yang berada pada modul *General Ledger*. Basis data pada sistem ini menggunakan sistem manajemen basis data *MySQL Cluster*. *Physical Data Model* (PDM) dari basis data sistem ini dijelaskan pada tabel 3.26 dan ditunjukkan pada Gambar 3.32.

Tabel 3. 55. Keterangan PDM

No	Tabel	Atribut	Keterangan
1	ap_transact ion	id : <i>integer</i>	Tabel untuk menyimpan data transaksi hutang perusahaan.
		ap_transaction_type: <i>integer</i>	
		transactionNumber : <i>varchar</i>	
		transactionName : <i>varchar</i>	
		date : <i>date</i>	
		dueDate : <i>date</i>	
		dueDateExt : <i>date</i>	
		amount : <i>integer</i>	
		total : <i>integer</i>	
		tax : <i>integer</i>	
		discount : <i>integer</i>	
		shipping : <i>integer</i>	

No	Tabel	Atribut	Keterangan
		additional_cost : <i>integer</i>	
		note : <i>varchar</i>	
		status : <i>varchar</i>	
		dunning : <i>integer</i>	
		supplier_id : <i>integer</i>	
		po_header_id : <i>integer</i>	
2	ap_paymen t	id : <i>integer</i>	Tabel untuk menyimpan data pembayaran hutang perusahaan
		paymentNumber :	
		<i>varchar</i>	
		paymentName : <i>varchar</i>	
		date : <i>date</i>	
		dueDate : <i>date</i>	
		amount : <i>integer</i>	
		residu : <i>integer</i>	
		status : <i>varchar</i>	
		note : <i>varchar</i>	
3	ap_paymen t_line	ap_transaction_id : <i>integer</i>	Tabel untuk menyimpan data pembayaran hutang perusahaan
		id : <i>integer</i>	
		payment_lineNumber :	
		<i>varchar</i>	
		payment_lineName :	
		<i>varchar</i>	
		pay : <i>integer</i>	
		note : <i>varchar</i>	
4	ar_transacti on	date : <i>date</i>	Tabel untuk menyimpan data transaksi piutang perusahaan.
		ap_payment_id : <i>integer</i>	
		gl_journal_id : <i>integer</i>	
		id : <i>integer</i>	
		ar_transaction_type:	
		<i>integer</i>	
		transactionNumber :	
		<i>varchar</i>	
		transactionName : <i>varchar</i>	
		date : <i>date</i>	

No	Tabel	Atribut	Keterangan
		dueDate : <i>date</i>	
		dueDateExt : <i>date</i>	
		amount : <i>integer</i>	
		total : <i>integer</i>	
		tax : <i>integer</i>	
		discount : <i>integer</i>	
		shipping : <i>integer</i>	
		additional_cost : <i>integer</i>	
		note : <i>varchar</i>	
		status : <i>varchar</i>	
		dunning : <i>integer</i>	
		sd_customer_id : <i>integer</i>	
		sd_sale_id : <i>integer</i>	
5	ar_paymen t	id : <i>integer</i>	Tabel untuk menyimpan data pembayaran piutang perusahaan
		paymentNumber :	
		<i>varchar</i>	
		paymentName : <i>varchar</i>	
		date : <i>date</i>	
		dueDate : <i>date</i>	
		amount : <i>integer</i>	
		residu : <i>integer</i>	
		status : <i>varchar</i>	
		note : <i>varchar</i>	
		ar_transaction_id : <i>integer</i>	
6	ar_paymen t_line	id : <i>integer</i>	Tabel untuk menyimpan data pembayaran piutang perusahaan.
		payment_lineNumber :	
		<i>varchar</i>	
		payment_lineName :	
		<i>varchar</i>	
		pay : <i>integer</i>	
		note : <i>varchar</i>	
		date : <i>date</i>	
		ar_payment_id : <i>integer</i>	
		gl_journal_id : <i>integer</i>	



Gambar 3.1. Physical Data Model Modul Account Payable dan Account Receivable

[Halaman ini sengaja dikosongkan]

3.6.3. Perancangan *RBAC (Role Base Access Control)*

Pada bagian ini akan dijelaskan mengenai perancangan *RBAC* yang terdapat pada sistem ERP 2016. Secara rinci mengenai *RBAC* dijabarkan sebagai berikut.

3.6.3.1. Perancangan antarmuka login

Antarmuka ini merupakan antarmuka yang digunakan untuk *login* atau masuk ke dalam sistem aplikasi. Terdapat form “*username*”, “*password*” dan “*setDB*”. Pengguna harus memasukkan *username* dan *password* yang telah terverifikasi. yang Rancangan antarmuka ini dapat dilihat pada Gambar 3.23.

The screenshot shows a web application interface for 'EZ ERP 2015'. At the top, there is a header bar with 'EZ ERP 2015' on the left and 'Login' on the right. Below the header is a navigation bar with 'Home / Login'. The main content area contains a 'Login' form. The form has three input fields: 'Username', 'Password', and 'Select DB' (a dropdown menu). A blue 'Login' button is located at the bottom of the form.

Gambar 3. 24. Perancangan antarmuka Login

3.6.3.2. Perancangan antarmuka *add user*

Antarmuka ini merupakan antarmuka yang digunakan untuk *add user* atau menambahkan pengguna baru ke dalam sistem aplikasi. Terdapat form “*username*”,

“email” dan “password”. Rancangan antarmuka ini dapat dilihat pada Gambar 3.24.

The image shows a web application interface. In the foreground, there is a modal window titled "Add New User" with a close button (X) in the top right corner. The modal contains three input fields: "Username", "Email", and "Password". Below these fields is a blue button. The background shows a sidebar with a "Home / Login" link and a top bar with "EZ ERP 2019" and a "Login" button.

Gambar 3. 25. Perancangan antarmuka *add user*

3.6.4. Perancangan Tampilan

Pada subbab ini akan dibahas mengenai perancangan antarmuka pengguna yang digunakan oleh sistem. Pembahasan perancangan antarmuka penulis bagi menjadi dua bagian yaitu perancangan antarmuka untuk Modul *Account Payable* dan *Account Receivable*

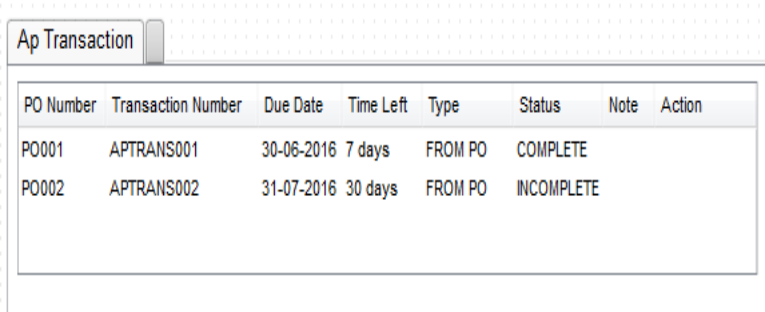
3.6.4.1. Perancangan Antarmuka untuk *Account Payable*

Pada modul *Account Payable* terdapat beberapa antarmuka. Masing-masing penjelasan mengenai antarmuka-antarmuka tersebut adalah sebagai berikut.

3.6.4.1.1. Perancangan Antarmuka Daftar *Account Payable Transaction*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat daftar *Account Payable Transaction* yang ada pada sistem. Terdapat sebuah tabel yang menampilkan informasi umum

mengenai setiap *Account Payable Transaction*. Pada setiap baris tabel terdapat pilihan menghapus, menyunting, dan melihat rincian *Account Payable Transaction* tertentu. Terdapat pula sebuah tombol “*Create Ap Transaction*” untuk mengarahkan pengguna ke antarmuka membuat *Account Payable Transaction*. Rancangan antarmuka ini dapat dilihat pada Gambar 3.25.



The screenshot shows a window titled "Ap Transaction" with a table containing two rows of transaction data. The table has columns for PO Number, Transaction Number, Due Date, Time Left, Type, Status, Note, and Action.

PO Number	Transaction Number	Due Date	Time Left	Type	Status	Note	Action
PO001	APTRANS001	30-06-2016	7 days	FROM PO	COMPLETE		
PO002	APTRANS002	31-07-2016	30 days	FROM PO	INCOMPLETE		

Gambar 3. 26. Rancangan Antarmuka Daftar *Account Payable Transaction*

3.6.4.1.2. Perancangan Antarmuka Membuat *Account Payable Transaction*

Antarmuka ini merupakan antarmuka yang digunakan untuk membuat *Account Payable Transaction*. Pada antarmuka terdapat beberapa TextBox untuk diisi berkaitan dengan informasi *Account Payable Transaction*. Terdapat juga sebuah *DropDownList* untuk memilih inputan *Purchase Oder Number*. Selanjutnya setelah semua data terisi, pengguna dapat menekan tombol “*Create*” untuk menyimpan data. Rancangan antarmuka ini dapat dilihat pada Gambar 3.26.

Create Ap Transaction

Ap Transaction Type

Purchase Order ID Currency ID Supplier ID

Transaction Name Amount

Date Tax

Due Date Discount

Note Total

Status

Create

Gambar 3. 27. Rancangan Antarmuka Membuat *Account Payable Transaction*

3.6.4.1.3. Perancangan Antarmuka Menyunting *Account Payable Transaction*

Antarmuka ini merupakan antarmuka yang digunakan untuk menyunting data *Account Payable Transaction* tertentu. Isian data pada antarmuka sama seperti pada antarmuka membuat *Account Payable Transaction*. Terdapat tombol “*Update*” yang berfungsi menyimpan hasil suntingan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.27.

Update Ap Transaction

Ap Transaction Type
From PO

Purchase Order ID: PO001 Currency ID: IDR Supplier ID: PT Sanjaya

Transaction Name: Transaction 1 Amount: 30000

Date: 10-06-2016 Tax: 3000

Due Date: 30-06-2016 Discount:

Note: Total: 33000

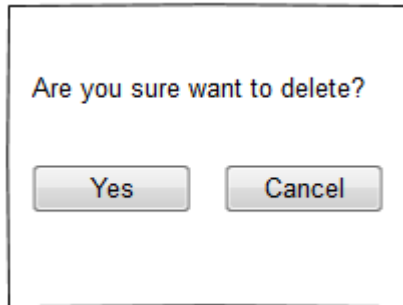
Status: INCOMPLETE

Update

Gambar 3. 28. Rancangan Antarmuka Menyunting *Account Payable Transaction*

3.6.4.1.4. Perancangan Antarmuka Menghapus *Account Payable Transaction*

Antarmuka ini berupa halaman *pop-up* yang meminta konfirmasi penghapusan *Account Payable Transaction*. Terdapat dua buah tombol yaitu tombol “Yes” untuk konfirmasi dan “Cancel” untuk pembatalan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.28.



Gambar 3. 29. Rancangan Antarmuka Menghapus *Account Payable Transaction*

3.6.4.1.5. Perancangan Antarmuka Daftar *Account Payable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat daftar *Account Payable Payment* yang ada pada sistem. Terdapat sebuah tabel yang menampilkan informasi umum mengenai setiap *Account Payable Payment*. Pada setiap baris tabel terdapat pilihan menghapus, menyunting, dan melihat rincian *Account Payable Payment* tertentu. Terdapat pula sebuah tombol “*Create Ap Payment*” untuk mengarahkan pengguna ke antarmuka Membuat *Account Payable Payment*. Rancangan antarmuka ini dapat dilihat pada Gambar 3.29.

Ap Payment									
Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status	Note	Action
APTRANS001	APPAYMENT001	30-06-2016	7 days	100000	100000	0	COMPLETE		
APTRANS002	APPAYMENT002	31-07-2016	30 days	7000	2000	5000	INCOMPLETE		

Gambar 3. 30. Rancangan Antarmuka Daftar *Account Payable Payment*

3.6.4.1.6. Perancangan Antarmuka Membuat *Account Payable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk membuat *Account Payable Payment*. Pada antarmuka terdapat beberapa *TextBox* untuk diisi berkaitan dengan informasi mengenai *Account Payable Payment*. Terdapat dua buah *DropDownList* untuk memilih inputan *Ap Transaction Number*. Selanjutnya setelah semua data terisi, pengguna dapat menekan tombol “*Create*” untuk menyimpan data. Rancangan antarmuka ini dapat dilihat pada Gambar 3.30.

Create Ap Payment					
Ap Transaction ID	Currency ID	Supplier ID			
<input type="text"/>	<input type="text"/>	<input type="text"/>			
Payment Name	<input type="text"/>	Amount	<input type="text"/>		
Date	<input type="text"/>	Paid	<input type="text"/>		
Due Date	<input type="text"/>	Residu	<input type="text"/>		
Date Complete	<input type="text"/>	Status	<input type="text"/>		
Note	<input type="text"/>				
Ap Payment Line					
Date Pay	Number	Name	Pay	GL Journal	Note
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Add"/>					
<input type="button" value="Create"/>					

Gambar 3. 31. Rancangan Antarmuka Membuat *Account Payable Payment*

3.6.4.1.7. Perancangan Antarmuka Menyunting *Account Payable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk menyunting data *Account Payable Payment* tertentu. Isian data pada antarmuka sama seperti pada antarmuka membuat *Account Payable Payment*. Terdapat tombol “*Update*” yang berfungsi menyimpan hasil suntingan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.31.

Update Ap Payment

Ap Transaction ID	Currency ID	Supplier ID
APTRANS001	IDR	PT Sanjaya
Payment Name	Payment 001	Amount
		1000000
Date	02-06-2016	Paid
		700000
Due Date	30-06-2016	Residu
		300000
Date Complete		Status
		INCOMPLETE
Note		

Ap Payment Line

Date Pay	Number	Name	Pay	GL Journal	Note
02-06-2016	Pay1	DP	700000	Journal 1	

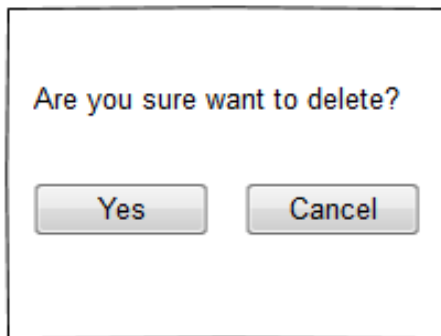
Add

Update

Gambar 3. 32. Rancangan Antarmuka Menyunting *Account Payable Payment*

3.6.4.1.8. Perancangan Antarmuka Menghapus *Account Payable Payment*

Antarmuka ini berupa halaman *pop-up* yang meminta konfirmasi penghapusan *Account Payable Payment*. Terdapat dua buah tombol yaitu tombol “*Yes*” untuk konfirmasi dan “*Cancel*” untuk pembatalan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.32.



Gambar 3. 33. Rancangan Antarmuka Menghapus *Account Payable Payment*

3.6.4.1.9. Perancangan Antarmuka Melihat *Account Payable Aging Analysis*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat umur hutang perusahaan. Pada antarmuka terdapat tabel yang berisi hutang perusahaan yang telah melewati tanggal jatuh tempo. Hutang tersebut dikelompokkan dalam beberapa periode, yaitu periode tiap bulan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.33.

Ap Aging Analysis									
Transaction Number	Amount	Due Date	Current	0-7 Days	8-14 Days	15-21 Days	22-28 Days	28+ Days	Total
PT Sanjaya									
APTRANS001	20000	30-06-2016						20000	20000
APTRANS002	10000	31-07-2016			10000				10000
								Total	30000

Gambar 3. 34. Rancangan Antarmuka Melihat *Account Payable Aging Analysis*

3.6.4.1.10 Perancangan Antarmuka Melihat *Account Payable Debt Analysis*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat urutan hutang perusahaan menurut tingkat kepentingannya. Pada antarmuka terdapat tabel yang berisi hutang perusahaan yang telah dilakukan proses perankingan sesuai skor terendah. Rancangan antarmuka ini dapat dilihat pada Gambar 3.34.

Ap Debt Analysis								
Transaction Number	Due Date	Fine	Debt	Tolerance	Late	Discount	Score	
APTRANS001	30-06-2016	100	1000	7	1	30	0.011	
APTRANS002	31-07-2016	50	2000	4	2	10	0.045	

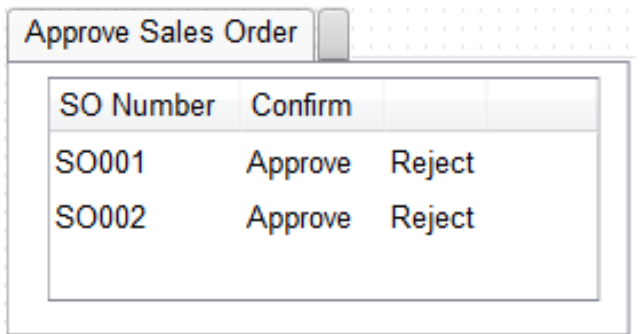
Gambar 3. 35. Rancangan Antarmuka Melihat *Account Payable Debt Analysis*

3.6.4.2. Perancangan Antarmuka untuk *Account Receivable*

Pada modul Account Payable terdapat beberapa antarmuka. Masing-masing penjelasan mengenai antarmuka-antarmuka tersebut adalah sebagai berikut.

3.6.4.2.1. Perancangan Antarmuka Menyetujui *Sales Order*

Antarmuka ini berupa halaman yang menampilkan daftar sales order yang telah disetujui. Terdapat dua buah tombol yaitu tombol “*Approve*” untuk persetujuan dan “*Reject*” untuk pembatalan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.35.



Gambar 3. 36. Rancangan Antarmuka Melakukan Konfirmasi *Sales Order*

3.6.4.2.2. Perancangan Antarmuka Daftar *Account Receivable Transaction*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat daftar *Account Receivable Transaction* yang ada pada sistem. Terdapat sebuah tabel yang menampilkan informasi umum mengenai setiap *Account Receivable Transaction*. Pada setiap baris tabel terdapat pilihan menghapus, menyunting, dan melihat rincian *Account Receivable Transaction* tertentu. Terdapat pula sebuah tombol “*Create New*” untuk mengarahkan pengguna ke antarmuka Membuat *Account Receivable*. Rancangan antarmuka ini dapat dilihat pada Gambar 3.36.

Ar Transaction							
SO Number	Transaction Number	Due Date	Time Left	Type	Status	Note	Action
SO001	APTRANS001	30-06-2016	7 days	FROM SO	COMPLETE		
SO002	APTRANS002	31-07-2016	30 days	FROM SO	INCOMPLETE		

Gambar 3. 37. Rancangan Antarmuka Daftar *Account Receivable Transaction*

3.6.4.2.3. Perancangan Antarmuka Membuat *Account Receivable Transaction*

Antarmuka ini merupakan antarmuka yang digunakan untuk membuat *Account Receivable Transaction*. Pada antarmuka terdapat beberapa TextBox untuk diisi berkaitan dengan informasi *Account Receivable Transaction*. Terdapat juga sebuah *DropDownList* untuk memilih inputan *Purchase Oder Number*. Selanjutnya setelah semua data terisi, pengguna dapat menekan tombol “*Create*” untuk menyimpan data. Rancangan antarmuka ini dapat dilihat pada Gambar 3.37.

Create Ar Transaction

Ar Transaction Type

Sales Order ID

Currency ID

Customer ID

Transaction Name

Date

Due Date

Note

Amount

Tax

Discount

Total

Status

Create

Gambar 3. 38. Rancangan Antarmuka Membuat *Account Receivable Transaction*

3.6.4.2.4. Perancangan Antarmuka Menyunting *Account Receivable Transaction*

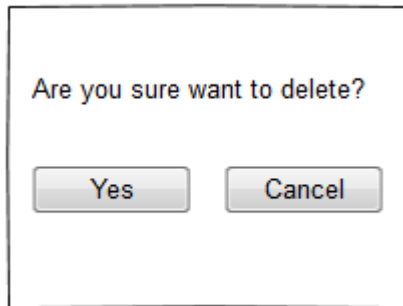
Antarmuka ini merupakan antarmuka yang digunakan untuk menyunting data *Account Receivable Transaction* tertentu. Isian data pada antarmuka sama seperti pada antarmuka membuat *Account Receivable Transaction*. Terdapat tombol “*Update*” yang berfungsi menyimpan hasil suntingan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.38.

Update Ar Transaction	
Ar Transaction Type From SO	
Sales Order ID	Currency ID
SO001	IDR
Customer ID	PT Sanjaya
Transaction Name	Transaction 1
Amount	30000
Date	10-06-2016
Tax	3000
Due Date	30-06-2016
Discount	
Note	
Total	33000
Status	INCOMPLETE
Update	

Gambar 3. 39. Rancangan Antarmuka Menyunting *Account Receivable Transaction*

3.6.4.2.5. Perancangan Antarmuka Menghapus *Account Receivable Transaction*

Antarmuka ini berupa halaman *pop-up* yang meminta konfirmasi penghapusan *Account Receivable Transaction*. Terdapat dua buah tombol yaitu tombol “Yes” untuk konfirmasi dan “Cancel” untuk pembatalan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.39.



Gambar 3. 40. Rancangan Antarmuka Menghapus *Account Receivable*

3.6.4.2.6. Perancangan Antarmuka Daftar *Account Receivable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat daftar *Account Receivable Payment* yang ada pada sistem. Terdapat sebuah tabel yang menampilkan informasi umum mengenai setiap *Account Receivable Payment*. Pada setiap baris tabel terdapat pilihan menghapus, menyunting, dan melihat rincian *Account Receivable Payment* tertentu. Terdapat pula sebuah tombol “Create New” untuk mengarahkan pengguna ke antarmuka Membuat *Account Receivable Payment*. Rancangan antarmuka ini dapat dilihat pada Gambar 3.40.

Ar Payment									
Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status	Note	Action
ARTRANS001	APPAYMENT001	30-06-2016	7 days	100000	100000	0	COMPLETE		
ARTRANS002	APPAYMENT002	31-07-2016	30 days	7000	2000	5000	INCOMPLETE		

Gambar 3. 41. Rancangan Antarmuka Daftar *Account Receivable Payment*

3.6.4.2.7. Perancangan Antarmuka Membuat *Account Receivable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk membuat *Account Receivable Payment*. Pada antarmuka terdapat beberapa TextBox untuk diisi berkaitan dengan informasi mengenai *Account Payable Payment*. Terdapat dua buah *DropDownList* untuk memilih inputan *Ar Transaction Number*. Selanjutnya setelah semua data terisi, pengguna dapat menekan tombol “*Create*” untuk menyimpan data. Rancangan antarmuka ini dapat dilihat pada Gambar 3.41.

Create Ar Payment

Ar Transaction ID

Currency ID

Customer ID

Payment Name

Amount

Date

Paid

Due Date

Residu

Date Complete

Status

Note

Ar Payment Line

Date Pay

Number

Name

Pay

GL Journal

Note

Add

Create

Gambar 3. 42. Rancangan Antarmuka Membuat *Account Receivable Payment*

3.6.4.2.8. Perancangan Antarmuka Menyunting *Account Receivable Payment*

Antarmuka ini merupakan antarmuka yang digunakan untuk menyunting data *Account Receivable Payment* tertentu. Isian data pada antarmuka sama seperti pada antarmuka membuat *Account Receivable Payment*. Terdapat tombol “*Update*” yang berfungsi menyimpan hasil suntingan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.42.

Update Ar Payment

Ar Transaction ID	Currency ID	Customer ID
APTRANS001	IDR	Sanjaya
Payment Name	Payment 001	Amount
		1000000
Date	02-06-2016	Paid
		700000
Due Date	30-06-2016	Residu
		300000
Date Complete		Status
		INCOMPLETE
Note		

Ar Payment Line

Date Pay	Number	Name	Pay	GL Journal	Note
02-06-2016	Pay1	DP	700000	Journal 1	

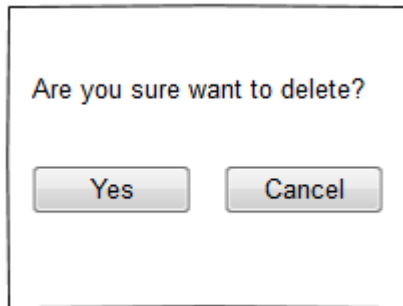
Add

Update

Gambar 3. 43. Rancangan Antarmuka Menyunting *Account Receivable Payment*

3.6.4.2.9. Perancangan Antarmuka Menghapus *Account Receivable Payment*

Antarmuka ini berupa halaman *pop-up* yang meminta konfirmasi penghapusan *Account Receivable Payment*. Terdapat dua buah tombol yaitu tombol “*Yes*” untuk konfirmasi dan “*Cancel*” untuk pembatalan. Rancangan antarmuka ini dapat dilihat pada Gambar 3.43.



Gambar 3. 44. Rancangan Antarmuka Menghapus *Account Receivable Payment*

3.6.4.2.10. Antarmuka Melihat *Account Receivable Aging Analysis*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat umur piutang perusahaan. Pada antarmuka terdapat tabel yang berisi piutang perusahaan yang telah melewati tanggal jatuh tempo. Piutang tersebut dikelompokkan dalam beberapa periode Rancangan antarmuka ini dapat dilihat pada Gambar 3.44.

Ar Aging Analysis									
Transaction Number	Amount	Due Date	Current	0-7 Days	8-14 Days	15-21 Days	22-28 Days	28+ Days	Total
Sanjaya									
APTRANS001	20000	30-06-2016						20000	20000
APTRANS002	10000	31-07-2016			10000				10000
								Total	30000

Gambar 3. 45. Rancangan Antarmuka Melihat *Account Receivable Aging Analysis*

3.6.4.2.11. Perancangan Antarmuka Melihat *Account Receivable Customer Analysis*

Antarmuka ini merupakan antarmuka yang digunakan untuk melihat urutan pelanggan yang telah melakukan transaksi. Pada antarmuka terdapat tabel yang berisi daftar urutan pelanggan yang telah melalui proses perankingan sesuai skor tertinggi. Rancangan antarmuka ini dapat dilihat pada Gambar 3.45.

Ar Customer Analysis							
Customer Number	Due Date	Fine	Debt	Tolerance	Late	Discount	Score
Cust 001	30-06-2016	100	1000	7	1	30	0.011
Cust 002002	31-07-2016	50	2000	4	2	10	0.045

Gambar 3. 46. Rancangan Antarmuka Melihat *Account Receivable Customer Analysis*

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem ERP 2015. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem serta proses implementasi *distributed database*, *RBAC*, *multi-tenancy*, dan antarmuka pengguna.

4.1. Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Database yang digunakan pada server adalah SQL Server 2012 R2.
2. PC untuk server menggunakan AMD A6-3400M APU with Radeon HD Graphics @1.40Ghz, RAM 4GB dengan Sistem Operasi Windows 7 Ultimate.
3. StarUML untuk pembuatan diagram, Power Designer 12.5, Sublime sebagai teks editor, Pencil untuk pembuatan desain antarmuka, Power Designer untuk pembuatan CDM dan PDM.
4. Mozilla Firefox 46.0.1 sebagai antarmuka untuk pengujian aplikasi klien.

4.2. Implementasi Basis Data Terdistribusi

Pada bagian ini akan dijelaskan mengenai implementasi *distributed database* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *distributed database* dijabarkan sebagai berikut:

4.2.1. Instalasi Data dan SQL node pada node1 dan node2

Pada implementasi instalasi dan sql node pada node1 dan node2, dilakukan langkah-langkah sebagai berikut:

1. Membuat grup MySQL pengguna baru, kemudian menambah user MySQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.1.

```
shell> groupadd mysql
shell> useradd-g mysql mysql
```

Kode Sumber 4.1. Membuat grup MySQL pengguna baru dan menambah user MySQL

2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, kemudian mengubah arsip dan menciptakan symlink ke dalam direktori mysql yang bernama “mysql”. Hal yang perlu diperhatikan adalah, file yang sebenarnya dan nama direktori bervariasi sesuai dengan jumlah cluster versi MySQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.2.

```
shell> cd/usr/local
shell>/usr/local$ tar xzvf mysql-cluster-gpl-7.1.34-
linux-x86_64-glibc23
shell>ln-s/usr/local/mysql-cluster-gpl-7.1.34-linux-x86-
glibc23/usr/local/mysql
shell> export PATH = $ PATH:/usr/local/mysql/bin
shell> echo "export PATH=\$PATH:/usr/local/ mysql/bin">>
/etc/bash.bashrc
```

Kode Sumber 4.2. Mengubah Lokasi Direktori, Mengubah Arsip, dan Menciptakan Symlink

3. Mengubah lokasi ke dalam direktori mysql dan menjalankan *script* untuk menciptakan *database system*. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.3.

```
shell> cd mysql
shell>. / scripts / mysql install db-user = mysql
```

Kode Sumber 4.3. Mengubah Lokasi Direktori

4. Mengatur izin yang diperlukan oleh server MSQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.4.

```
shell> chown-R root.
shell> chown-R mysql data
shell> chgrp-R mysql.
```

Kode Sumber 4.4. Mengatur Perizinan Server MySQL

5. Menyalin *script startup MySQL* ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.5.

```
shell> cp support-files/mysql.server / etc / init.d /
mysql
shell> chmod + x / etc / init.d / mysql
shell> update-rc.d mysql defaults
```

Kode Sumber 4.5. Menyalin script startup MySQL

4.2.2. Pemasangan Manajemen Node pada node03

Pemasangan node manajemen memerlukan manajemen server MySQL Cluster (ndb_mgmd), diasumsikan bahwa mysql-cluster-gpl-7.1.5-linux-i686-glibc23.tar.gz telah ditempatkan di /var / tmp. Untuk memasang ndb_mgmd dan ndb_mgm pada host Cluster, sistem sebagai root melakukan langkah-langkah sebagai berikut:

1. Mengubah lokasi ke dalam direktori / var / tmp direktori, dan mengekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.6.

```
shell> cd/usr/local
shell> tar-zxvf mysql-cluster-gpl-7.1.34-linux-x86-
glibc23.tar.gz
shell> cd / usr/local/mysql-cluster-gpl-7.1.34-
linux-x86-glibc235
shell> cp bin / ndb_mgm */usr/local/bin
```

Kode Sumber 4. 6. Mengubah Lokasi Direktori

2. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.7.

```
shell> cd/usr/local/bin
shell> chmod +x ndb mgm*
```

Kode Sumber 4. 7. Mengubah Lokasi Direktori

4.2.3. Konfigurasi Manajemen Node

Konfigurasi pada manajemen node dilakukan dengan langkah-langkah sebagai berikut:

1. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.8.

```
shell> mkdir / var / lib / mysql-cluster
shell> cd / var / lib / mysql-cluster
vi config.ini
```

Kode Sumber 4. 8. Membuat Direktori

2. Mengatur file “config.ini”. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.9.

```
[ndbd default]
NoOfReplicas=2
DataMemory=10G
IndexMemory=2G
MaxNoOfAttributes=10000
[tcp default]
[ndb_mgmd]
hostname=10.151.64.182
datadir=/var/lib/mysql-cluster
[ndbd]
hostname=10.151.64.182
datadir=/usr/local/mysql/data
[ndbd]hostname=10.151.64.203
datadir=/usr/local/mysql/data
[mysqld]MaxNoOfAttributes=10000
hostname=10.151.64.182
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.203
```

Kode Sumber 4. 9. Mengatur file "config.ini"

4.2.4. Konfigurasi Data dan SQL Node

Konfigurasi data dan SQL Node dilakukan dengan cara mengedit file `my.cnf` pada direktori `/etc/`. Untuk setiap data node dan SQL node yang diatur pada `my.cnf`. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.10.

```
[client]
port=3306
socket=/tmp/mysql.sock
[mysqld]
port=3306
socket=/tmp/mysql.sock
ndbcluster
ndb-connectstring=10.151.64.181
[mysql_cluster]
ndb-connectstring=10.151.64.181
```

Kode Sumber 4. 10. Data dan SQL Node

4.2.5. Memulai MySQL Cluster

Setiap proses node cluster harus dimulai secara terpisah. Manajemen node harus dimulai terlebih dahulu, kemudian node data. Pada setiap node SQL dilakukan langkah sebagai berikut :

1. Pada node03 (host manajemen), untuk memulai proses manajemen node dari shell sistem dilakukan perintah berikut. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.11.

```
shell> ndb_mgmd -f /var/lib/mysql-
cluster/config.ini-configdir=/var/lib/mysql-
cluster/
```

Kode Sumber 4. 11. Memulai Proses Manajemen Node

2. Jalankan perintah untuk memulai `ndbd` dan proses `mysql` server pada masing-masing Data/host SQL. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.12.

```
shell> /usr/local/mysql/bin/ndbd
```

Kode Sumber 4. 12. Memulai Proses `ndbd` dan MySQL Server

3. Mengaktifkan mysql pada data node. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.13.

```
shell> / etc / init.d / mysql start
```

Kode Sumber 4. 13. Mengaktifkan MySQL

4.3. Implementasi RBAC (Role Base Acces Control)

Pada bagian ini akan dijelaskan mengenai implementasi *RBAC* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *RBAC* dijabarkan sebagai berikut:

4.3.1. Membuat Tabel Pengguna

Pembuatan tabel pengguna dilakukan dengan menjalankan perintah “*yii migrate*” pada folder aplikasi. Jika berhasil yang maka akan muncul keterangan “*migrate successfully*”.

4.3.2. Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna

Pada tahap ini dibutuhkan 4 tabel autentifikasi yang terdiri dari:

1. Tabel Item : tabel yang menyimpan daftar otorisasi, standar nama tabel adalah *auth_item*.
2. Tabel Child : tabel yang menyimpan hirarki daftar otoristas, standar nama tabel adalah *auth_item_child*.
3. Tabel Assignment : , tabel yang menyimpan penetapan user untuk daftar otorisasi, standar nama tabel adalah *auth_assignment*.
4. Tabel Rule : tabel yang menyimpan aturan-aturan autentifikasi, standar nama tabel adalah *auth_rule*

Cara untuk membuat tabel-tabel tersebut yaitu dengan menjalankan perintah yang ditunjukkan pada Kode Sumber 4.14.

```
Yii migrate --migrationPath=@yii/rbac/migrations
```

Kode Sumber 4.14. Generate Tabel Autentifikasi

Jika berhasil yang maka akan terbuat 4 tabel pada database, yaitu tabel *auth_assignment*, tabel *auth_item*, tabel *auth_item_child* dan tabel *auth_rule*.

4.3.3. Membuat Modul Admin

Proses ini bertujuan untuk meletakkan konfigurasi pengguna dan masing-masing model dari 4 tabel autentifikasi. Kemudian dilakukan konfigurasi pada folder config file *web.php*. Terdapat 3 konfigurasi yaitu:

1. Admin : digunakan sebagai akses ke modul admin
2. Auth Manager : sebagai autentifikasi di yii2 dan mengatur role default sebagai *guest*
3. Session Time Out : mengatur durasi *time out session* selama 5 menit atau 300 detik.

Masing –masing konfigurasi tersebut ditunjukkan pada Kode Sumber 4.15.

```
'admin' => [ 'class' => 'app\modules\admin\AdminModule', ],
'components' => [ 'authManager' => [ 'class' =>
'yii\rbac\DbManager', 'defaultRoles' => ['guest'], ],
'session' => [ 'timeout' => 300, ],
```

Kode Sumber 4. 15. Pembuatan Modul Admin

4.3.4. Membuat Model Tabel Autentifikasi, *Controller* dan *View* Pengguna

Pada tahap ini dilakukan *generate* kelas model dari masing-masing tabel autentifikasi pada modul admin yang diperlukan pada tahap sebelumnya. Kemudian ditambahkan *generate* kelas *controller* dan *view* pada tabel pengguna. Proses *generate* ini menggunakan *yii generator* yang telah disediakan oleh framework yii.

4.3.5. Menambahkan Kode pada Kelas *usercontroller*

Pada modul admin, file *usercontroller.php* ditambahkan kode fungsi untuk semua tabel autentifikasi. Masing-

masing fungsi ditunjukkan pada Kode Sumber 4.16, Kode Sumber 4.17, Kode Sumber 4.18.

```
public function actionAuthItem()
{
    $auth = Yii::$app->authManager;
    // menambahkan akses sebagai admin ke tabel auth_item
    $admin = $auth->createPermission('admin');
    $admin->description = 'Allow user to access all page';
    $auth->add($admin);
    // menambahkan akses sebagai asset management manager
    ke tabel auth_item
    $am_manager = $auth->createPermission('am-manager');
    $am_manager->description = 'Allow user as Asset
    Management Manager';
    $auth->add($am_manager);
    ...
    // menambahkan akses sebagai sales and distribution
    staff ke tabel auth_item
    $sd_staff = $auth->createPermission('sd-staff');
    $sd_staff->description = 'Allow user as Sales and
    Distribution Staff';
    $auth->add($sd_staff);
}
```

Kode Sumber 4. 16. Kode Fungsi Tabel AuthItem

```
public function actionItemChild(){
    $auth = Yii::$app->authManager;

    //admin dapat mengakses semua daftar izin akses
    $ap_manager = $auth->createPermission('ap-manager');
    $ap_staff = $auth->createPermission('ap-staff');
    $ar_manager = $auth->createPermission('ar-manager');
    $ar_staff = $auth->createPermission('ar-staff');

    $admin = $auth->createRole('admin');
    $auth->add($admin);
    $auth->addChild($admin, $ap_manager);
    $auth->addChild($admin, $ap_staff);
    $auth->addChild($admin, $ar_manager);
    $auth->addChild($admin, $ar_staff);
}
```

Kode Sumber 4. 17. Kode Fungsi Tabel ItemChild

```

public function actionAuthAssignment() {
    $auth = Yii::$app->authManager;

    $admin = $auth->createRole('admin');

    $auth->assign($admin, 1);
}

```

Kode Sumber 4. 18. Kode Fungsi Tabel AuthAssignment

4.4. Implementasi Multitenancy

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari pengambilan data partisipan, hingga proses peningkatan level dan penghentian pada setiap *training*. Secara rinci mengenai implementasi RBAC dijabarkan sebagai berikut:

4.4.1. Membuat Halaman Muka Tenant

Pada implementasi membuat halaman tenant ini dilakukan pembuatan halaman tenant secara sederhana, kemudian ditambahkan pembuatan database untuk setiap tenant. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.19.

```

// mendapatkan nilai yang dimasukkan dari view
$host = "10.151.64.182";
$tenant = Yii::$app->request->post('tenant');
$database = Yii::$app->request->post('database');

// membuka koneksi pada untuk memasukkan data tenant
$koneksidb = mysqli_connect($host, "root", "",
    "multitenant");
if ($koneksidb->connect_error) {
    die("Connection failed: " . $koneksidb->connect_error);
}
// query untuk memasukkan data tenant
$insert = "insert into tenant
(tenant,database_name,created_at)
('".$tenant."','".$database."',NOW())";

if($tenant != "" && $database != ""){
    $koneksidb->query($insert);
}

```

```
}
// memutuskan koneksi ke server
    $koneksiidb->close();
```

Kode Sumber 4. 19. Pembuatan Halaman Muka Tenant

4.4.2. Menambahkan Database untuk Tenant Baru

Pada tahap ini dilakukan penambahan database untuk tenant baru dengan cara melakukan konfigurasi pada server node mysql. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.20

```
'mysql -u root -p;' lalu tekan enter
'create database <nama database>';' lalu tekan enter
'use <nama database>;' lalu tekan enter
e /tmp/mysql-dump/final-db.sql
```

Kode Sumber 4. 20. Penambahan Database Tenant Baru

4.4.3. Login Tenant

Setelah proses pembuatan database dan replikasi pada tahap 4.4.2. selesai, maka tenant melakukan login dengan memilih salah satu database, nama database yang dipilih tersebut disimpan dalam session dan akan digunakan untuk koneksi yang akan dibuat. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.21.

```
$host = '10.151.64.182'; // node mysql $port = 3306;
$waitTimeoutInSeconds = 1;
$src =
    @fsockopen($host,$port,$errCode,$errStr,$waitTimeoutI
nSeconds);
if(is_resource($src)){
    $_SESSION['dbserver_ip'] = "10.151.64.182"; // node mysql
    // penyimpanan nama database sebagai session untuk
    multitenancy
    $dbname = isset($_SESSION['database_name']) ?
        $_SESSION['database_name'] : 'test';
} else { $_SESSION['dbserver_ip'] = "10.151.64.203"; //
    node mysql
}
    $connection = [
        'class' => 'yii\db\Connection',
        'dsn' =>
            'mysql:host='.$_SESSION['dbserver_ip'].';dbname='.$db
name.'
```

```
'username' => 'root',
'password' => '',
'charset' => 'utf8'];
return $connection;
```

Kode Sumber 4. 21. Login Tenant

4.5. Implementasi Program pada Modul Account Payable

Pada bagian ini akan dijelaskan mengenai implementasi yang terdapat pada Modul *Account Payable* yang terbagi menjadi tampilan halaman utama atau *dashboard* dan beberapa sub modul, antara lain: *Account Payable Transaction*, *Account Payable Payment*, *Account Payable Aging Analysis*, *Account Payable Debt Analysis*. Secara rinci mengenai implementasi lapisan antarmuka Modul *Account Payable* dijabarkan sebagai berikut:

4.5.1. Halaman Utama Modul Account Payable

Halaman utama modul *Account Payable* menampilkan ringkasan informasi dari beberapa sub modul yang ada pada *Account Payable*. Pengaturan tampilan tersebut diatur pada kelas *DefaultController* dalam fungsi *actionIndex()* . Kode tampilan yang dimaksud ditunjukkan oleh Kode Sumber 4.22.

```
class DefaultController extends Controller
{
    public function actionIndex($item = "1")
    {
        $transaction = 0;
        $payment = 0;
        $transaction = ApTransaction::find()->
            where('created_at > DATE_SUB(NOW(),
            INTERVAL 1 DAY)')->count();
        $transactioncomplete = ApTransaction::find()->
            where('ap_transaction_status_id = 2')
            ->All();
        $transactionincomplete = ApTransaction::find()->
            where('ap_transaction_status_id = 1'
            )->All();
        $counttransactioncomplete = count($transactioncomplete);
        $counttransactionincomplete=count($transactionincomplete);
        $shutang = ApTransaction::find()->
            select('ap_transaction.date as date,
            ap_transaction.total as total')
            ->orderBy('date')->asArray()->All();
```

```

        $paymentcomplete = ApPayment::find()->where('status =
2')->All();
        $paymentincomplete = ApPayment::find()-
>where('status = 1')->All();
        $countpaymentcomplete = count($paymentcomplete);
        $countpaymentincomplete = count($paymentincomplete);
        $debt = ApPayment::find()->sum('residu');
        $payment = ApPayment::find()->where('created_at >
DATE_SUB(NOW(), INTERVAL 1 DAY)')->count();
        return $this->render('index', ['hutang' => $hutang,
            'transaction' => $transaction,
'payment' => $payment,
            'counttransactioncomplete' =>
        $counttransactioncomplete,
            'counttransactionincomplete' =>
        $counttransactionincomplete,
            'countpaymentcomplete' => $countpaymentcomplete,
            'countpaymentincomplete' =>
        $countpaymentincomplete,
            'debt' => $debt,
        ]);
    }
}

```

Kode Sumber 4. 22. Halaman Utama Modul *Account Payable*

4.5.2. Melihat Daftar *Account Payable Transaction*

Pada implementasi melihat daftar *Account Payable Transaction*, sistem menampilkan seluruh daftar transaksi yang berhasil ditambahkan pada *Account Payable Transaction*. Pengaturan tampilan ini diatur pada kelas *ApTransactionController* fungsi *actionIndex()* Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada Kode Sumber 4.23.

```

public function actionIndex()
{
    $searchModel = new ApTransactionSearch();
    $dataProvider = $searchModel->search(Yii::$app-
>request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

Kode Sumber 4.23. Melihat Daftar *Account Payable Transaction*

4.5.3. Menambah *Account Payable Transaction*

Pada implementasi menambah *Account Payable Transaction*, sistem menambahkan transaksi baru ke dalam *Account Payable Transaction*. Pengaturan menambah transaksi baru ini diatur pada kelas *ApTransactionController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada Kode Sumber 4.24.

```
public function actionCreate()
{
    $model = new ApTransaction();
    $model->date = date('Y-m-d');

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $poId = $model['pur_po_header_id'];
        $status =
    $model['ap_transaction_status'];
        PurPoHeaderController::actionApStatus($poId, "1");
        $poId = $model['pur_po_header_id'];
        $status =
    $model['ap_transaction_status_id'];
        if(!empty($poId)){
            PurPoHeaderController::actionApStatus($poId,
            "1");
        }
        $newId =
    str_pad($model['id'],11,"APTRANS0000",STR_PAD_LEFT);
        $data = ApTransaction::findOne(['id' =>
    $model['id']]);
        $data->transactionNumber = $newId;
        $data->ap_transaction_status_id ='2';
        $data->update();
        return $this->redirect(['view', 'id' => $model->
    id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}
```


Kode Sumber 4.24. Menambah *Account Payable Transaction*

Fungsi `actionCreate()` pada Kode Sumber 4.24. berfungsi untuk menambahkan transaksi baru dimana ketika transaksi berhasil ditambahkan maka akan mengubah status *Purchase Order* yang dipilih menjadi 1. Status tersebut menjadi penanda agar ketika akan menambah transaksi baru, *Purchase Order* yang sudah berstatus 1 tidak muncul kembali saat mengisi form menambah transaksi baru. Penamaan nomor transaksi dibuat otomatis dengan format 'APTRANS0000', 4 digit terakhir menyesuaikan dengan id transaksi. Status transaksi otomatis berstatus 2, artinya transaksi tersebut belum lunas atau 'incomplete'. Status transaksi tersebut akan berubah menjadi 1 atau 'complete' ketika pembayaran sudah dilakukan dan lunas.

4.5.4. Menyunting *Account Payable Transaction*

Pada implementasi menyunting *Account Payable Transaction*, sistem memperbarui data transaksi pada *Account Payable Transaction*. Pengaturan menyunting transaksi ini diatur pada kelas *ApTransactionController* fungsi `actionUpdate()`. Kode untuk menyunting *Account Payable Transaction* ditunjukkan pada Kode Sumber 4.25.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);
    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $poId = $model['pur_po_header_id'];
        $status = $model['ap_transaction_status'];
        $appayment =
    ApPayment::findOne(['ap_transaction_id' => $model['id']]);
        if (!empty($appayment)) {
            $appayment->amount = $model['total'];
            $appayment->update();
        }
        PurPoHeaderController::actionApStatus($poId,
    "1");
        $poId = $model['pur_po_header_id'];
        $status = $model['ap_transaction_status_id'];
        PurPoHeaderController::actionApStatus($poId,
```

```

"1");
        PurPoHeaderController::actionPaymentStatus($poId,
$status);
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4. 25. Menyunting *Account Payable Transaction*

Fungsi *actionUpdate()* pada Kode Sumber 4.25. berfungsi untuk memperbarui transaksi dimana ketika transaksi berhasil diperbarui maka juga akan mengubah nilai *amount* pada *Account Payable Payment* apabila terjadi perubahan nilai total pada *Account Payble Transaction*.

4.5.5. Menghapus *Account Payable Transaction*

Pada implementasi menghapus *Account Payable Transaction*, sistem menghapus data transaksi pada *Account Payable Transaction*. Pengaturan menyunting transaksi ini diatur pada kelas *ApTransactionController* fungsi *actionDelete()*. Kode untuk menghapus *Account Payable Transaction* ditunjukkan pada Kode Sumber 4.26.

```

public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();
    return $this->redirect(['index']);
}

```

Kode Sumber 4. 26. Menghapus *Account Payable Transaction*

4.5.6. Melihat Daftar *Account Payable Payment*

Pada implementasi melihat daftar *Account Payable Payment*, sistem menampilkan seluruh daftar pembayaran yang berhasil ditambahkan pada *Account Payable Payment*. Pengaturan tampilan ini diatur pada kelas *ApPaymentController* fungsi

actionIndex(). Kode untuk menampilkan seluruh pembayaran tersebut ditunjukkan pada Kode Sumber 4.27.

```
public function actionIndex()
{
    $searchModel = new ApPaymentSearch();
    $dataProvider = $searchModel->search(Yii::$app-
    >request->queryParams);
    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}
```

Kode Sumber 4. 27. Melihat Daftar Account Payable Payment

4.5.7. Menambah Account Payable Payment

Pada implementasi menambah *Account Payable Payment*, sistem menambahkan pembayaran baru ke dalam *Account Payable Payment*. Pengaturan menambah pembayaran baru ini diatur pada kelas *ApPaymentController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh pembayaran tersebut ditunjukkan pada Kode Sumber 4.28.

```
public function actionCreate()
{
    $model = new ApPayment();
    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()/* && $apmodel->saveAll()*/) {
        $newId =
        str_pad($model['id'],13,"APPAYMENT0000",STR_PAD_LEFT);
        $code = ApPayment::findOne(['id' =>
        $model['id']]);
        $code->paymentNumber = $newId;
        $code->update();
        if($model['ap_transaction_id'] != 0){
            $apmodel =
            ApTransaction::findOne($model['ap_transaction_id']);
            $poId = $apmodel['pur_po_header_id'];
            $apmodel['status_payment'] = '1';
            if($model['status'] == 'COMPLETE')
            {
                $apmodel['ap_transaction_status_id'] =
                '1';
            }
            PurPoHeaderController::actionPaymentStatus($poId, '1');
            PurRoHeaderController::actionPaymentStatus($poId, '1');
        }
        else
    }
```

```

        {
            $apmodel['ap_transaction_status_id'] =
'2';
PurPoHeaderController::actionPaymentStatus($poId, '2');
PurRoHeaderController::actionPaymentStatus($poId, '2');
        }
        $apmodel->update();
    }
    $ref = array();
    foreach ($model['apPaymentLines'] as $key =>
$value) {
        $data = array(
            'payment_no' =>
$value['payment_lineNumber'],
            'payment_name' =>
$value['payment_lineName'],
        );
        array_push($ref, $data);
    }
    foreach ($model['apPaymentLines'] as $key =>
$value) {
JournalHeaderController::actionInsert($value['gl_journal_id']
, $ref[$key], $value['Pay'], $poId, 'ap', 1,
$model['status']);
        JournalHeaderController::actionShip('ap',
$ref[$key], $value['ap_payment_line_category_id'], $poId);
    }
    return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}
}

```

Kode Sumber 4. 28. Menambah *Account Payable Payment*

Fungsi *actionCreate()* pada Kode Sumber 4.28. berfungsi untuk menambahkan pembayaran baru dimana ketika pembayaran berhasil ditambahkan maka akan mengubah status *Account Payable Transaction* yang dipilih menjadi 1. Status tersebut menjadi penanda agar ketika akan menambah pembayaran baru, *Account Payable Transaction* yang sudah berstatus 1 tidak muncul kembali saat mengisi form menambah pembayaran. Nomor transaksi dibuat otomatis dengan format 'APPAYMENT0000, 4

digit terakhir menyesuaikan dengan id pembayaran. Nominal pembayaran ditambahkan pada *tabular form ApPaymentLine*. Setiap nominal pembayaran akan masuk ke dalam *Journal Header*. Status pembayaran otomatis *'incomplete'*, status tersebut akan berubah menjadi *'complete'* apabila jumlah pembayaran sama dengan jumlah hutang. Status pembayaran ini terhubung dengan *Purchase Order* yang ada pada *Account Payable Transaction* yang dipilih, sehingga *purchasing order staff* juga dapat mengetahui status pembayaran dari masing-masing *Purchase Order* yang sudah mereka buat sebelumnya.

4.5.8. Menyunting Account Payable Payment

Pada implementasi menyunting *Account Payable Payment*, sistem memperbarui data pembayaran pada *Account Payable Payment*. Pengaturan menyunting transaksi ini diatur pada kelas *ApPaymentController* fungsi *actionUpdate()*. Kode untuk menyunting *Account Payable Payment* ditunjukkan pada Kode Sumber 4.29.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);
    if ($model->loadAll(Yii::$app->request->post())) {
        if($model['ap_transaction_id'] != 0){
            $apmodel =
ApTransaction::findOne($model['ap_transaction_id']);
            $poId = $apmodel['pur_po_header_id'];
            $apmodel['status_payment'] = '1';
            if($model['status'] == 'COMPLETE')
            {
                $apmodel['ap_transaction_status_id']=
'1';
PurPoHeaderController::actionPaymentStatus($poId, '1');
            }
            else
            {
                $apmodel['ap_transaction_status_id']=
'2';
PurPoHeaderController::actionPaymentStatus($poId, '2');
            }
            $apmodel->update();
        }
    }
}
```

```

        $ref = array();
        foreach($model['apPaymentLines'] as $key=>
$value) {
            $data = array(
                'payment_no'=>
$value['payment_lineNumber'],
                'payment_name'=>
$value['payment_lineName'],
            );
            array_push($ref, $data);
        }
        foreach($model['apPaymentLines'] as $key=> $value)
        {
            $cekLine = "SELECT * FROM ap_payment_line
WHERE id='$value[id]';";
            $db = Yii::$app->db;
            $cek = $db->createCommand($cekLine)-
>queryOne();
            if(!isset($cek['id']) || $cek['id'] == "")
            {
                JournalHeaderController::actionInsert($value['gl_journ
al_id'], $ref[$key], $value['Pay'], $poId, 'ap', 2);
            }
            $model->saveAll();
            return $this->redirect(['view', 'id' => $model-
>id]);
        } else {return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4. 29. Menyunting Account Payable Payment

Fungsi *actionUpdate()* pada Kode Sumber 4.29. berfungsi untuk memperbaiki pembayaran dimana ketika pembayaran berhasil diperbarui maka juga akan mengubah status pada *Purchase Order* apabila terjadi perubahan status pada *Account Payable Payment*. Fungsi ini juga akan memperbaiki *Journal Header* apabila terjadi penambahan nominal pembayaran pada *ApPaymentLine*.

4.5.9. Menghapus Account Payable Payment

Pada implementasi menghapus *Account Payable Payment*, sistem menghapus data pembayaran pada *Account Payable Payment*. Pengaturan menghapus pembayaran ini diatur pada kelas *ApPaymentController* fungsi *actionDelete()*. Kode untuk

menghapus *Account Payable Payment* ditunjukkan pada Kode Sumber 4.30.

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}
```

Kode Sumber 4. 30. Menghapus *Account Payable Payment*

4.5.10. Melihat *Account Payable Aging Analysis*

Pada implementasi melihat *Account Receivable Aging Analysis*, sistem menampilkan data hutang pelanggan yang telah melewati tanggal jatuh tempo. Pengaturan melihat umur hutang ini diatur pada kelas *ArAgingController* fungsi *actionIndex()*. Kode untuk menampilkan daftar umur hutang pelanggan ditunjukkan pada Kode Sumber 4.31.

```
public function actionIndex()
{
    $searchModel = new ApAgingSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    $query = "SELECT    currency.codeCurrency,
supplier.supplier_name , ap_transaction.transactionNumber,
ap_payment.residu as amount, ap_transaction.dueDate FROM
currency, ap_transaction, ap_payment, supplier WHERE
currency.id          = ap_transaction.currency_id      and
ap_payment.ap_transaction_id = ap_transaction.id      and
supplier.id          = ap_transaction.supplier_id      and
ap_transaction.ap_transaction_status_id = '2' ORDER BY
supplier.supplier_name";
    $db = Yii::$app->db;
    $command = $db->createCommand($query);
    $dataApAging = $command->queryAll();

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
        'dataApAging' => $dataApAging
    ]);
}
```

Kode Sumber 4. 31. Melihat *Account Payable Aging Analysis*

4.5.11. Melihat *Account Payable Debt Analysis*

Pada implementasi melihat *Account Receivable Customer Analysis*, sistem menampilkan data pelanggan yang pernah melakukan transaksi pembelian. Pengaturan melihat daftar pelanggan ini diatur pada kelas *ArCustomerController* fungsi *actionIndex()*. Kode untuk menampilkan daftar pelanggan ditunjukkan pada Kode Sumber 4.32.

```
public function actionIndex()
{
    $query = "SELECT      APT.transactionNumber,
APT.transactionName,      APT.dueDate,      APT.id, (SELECT
supplier.payment_fine FROM  ap_transaction, supplier WHERE
ap_transaction.supplier_id = supplier.id      AND
ap_transaction.id = APT.id)      AS      denda, (SELECT
ap_transaction.total_price FROM  ap_transaction, supplier
WHERE      ap_transaction.supplier_id = supplier.id      AND
ap_transaction.id = APT.id)      AS
total_price, (ap_payment.residu)      AS      banyak_hutang, (SELECT
supplier.payment_tolerance FROM  ap_transaction, supplier
WHERE      ap_transaction.supplier_id = supplier.id      AND
ap_transaction.id = APT.id)      AS      boleh_mundur, (SELECT
ap_transaction.dueDate FROM      ap_transaction      WHERE
ap_transaction.id = APT.id)      AS      time_left, (SELECT
COUNT(ap_transaction.transactionName) FROM  ap_transaction,
ap_dunning WHERE ap_transaction.ap_dunning_id = ap_dunning.id
AND ap_transaction.id = APT.id)      AS      frekuensi_telat, (SELECT
supplier.payment_discount FROM  supplier, ap_transaction WHERE
supplier.id = ap_transaction.supplier_id      AND
ap_transaction.id = APT.id)      AS      discount FROM  ap_transaction
AS      APT, ap_payment, supplier WHERE      APT.supplier_id =
supplier.id AND APT.id = ap_payment.ap_transaction_id GROUP
BY APT.id ORDER BY APT.ID ASC" ;

    $db = Yii::$app->db;

    $command = $db->createCommand($query);
    $results = $command->queryAll();
    $data = $results;

    return $this->render('index', [
        'data' => $data,
    ]);
}
```

Kode Sumber 4. 32. Melihat *Account Payable Debt Analysis*

4.6. Implementasi Program Modul *Account Receivable*

Dibagian ini akan dijelaskan mengenai implementasi yang terdapat pada Modul *Account Receivable* yang terbagi menjadi tampilan halaman utama atau *dashboard* dan beberapa sub modul, antara lain: *Account Receivable Transaction*, *Account Receivable Payment*, *Account Receivable Aging Analysis*, *Account Receivable Customer Analysis*. Secara rinci mengenai implementasi lapisan antarmuka Modul *Account Receivable* dijabarkan sebagai berikut:

4.6.1. Halaman Utama *Account Receivable*

Halaman utama modul *Account Receivable* menampilkan ringkasan informasi dari beberapa sub modul yang ada pada *Account Receivable*. Pengaturan tampilan tersebut diatur di kelas *DefaultController* dalam fungsi *actionIndex()*. Kode tampilan yang dimaksud ditunjukkan oleh Kode Sumber 4.33.

```
class DefaultController extends Controller
{
    public function actionIndex($item = "1")
    {
        $transaction = ArTransaction::find()->where('created_at > DATE_SUB(NOW(), INTERVAL 1 DAY)')->count();

        $counttransactioncomplete = count($transactioncomplete);

        $counttransactionincomplete = count($transactionincomplete);

        $paymentcomplete = ArPayment::find()->where('status = 2')->All();
        $paymentincomplete = ArPayment::find()->where('status = 1')->All();

        $countpaymentcomplete = count($paymentcomplete);
        $countpaymentincomplete = count($paymentincomplete);
        $debt = ArPayment::find()->sum('residu');
    }
}
```

Kode Sumber 4. 33. Halaman Utama *Account Receivable*

4.6.2. Melakukan Konfirmasi *Sales Order*

Pada implementasi menyetujui *Sales Order* sistem menampilkan daftar *sales order* kemudian mengubah status *sales order* menjadi approved apabila dipilih tombol *approve* dan menjadi 0 apabila dipilih tombol *reject*. Pengaturan tampilan ini diatur pada kelas *ArApproveController* fungsi *actionIndex()*. Kode tampilan yang dimaksud ditunjukkan oleh Kode Sumber 4.34.

```
public function actionIndex()
{
    $query = "SELECT sd_quotation.quotationstatus_sale as
status,          sd_quotation.id          as          id,
sd_quotation.quotationcode_sale          as          code,
sd_customer.name_customer          as          customer,
sd_quotation.net_sale          as          net,
sd_quotation.validto_sale as valid FROM sd_quotation,
sd_customer WHERE sd_quotation.sd_customer_id =
sd_customer.id AND (sd_quotation.quotationstatus_sale
= '0' OR sd_quotation.quotationstatus_sale = 2 OR
sd_quotation.quotationstatus_sale = 4)";

    $db = Yii::$app->db;
    $command = $db->createCommand($query);
    $result = $command->queryAll();

    return $this->render('index', ['dataProvider' =>
$result,
]);
}
```

Kode Sumber 4. 34. Melakukan Konfirmasi *Sales Order*

4.6.3. Melihat Daftar *Account Receivable Transaction*

Pada implementasi melihat daftar *Account Receivable Transaction*, sistem menampilkan seluruh daftar transaksi yang berhasil ditambahkan pada *Account Receiveble Transaction*. Pengaturan tampilan ini diatur pada kelas *ArTransactionController* fungsi *actionIndex()* Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada Kode Sumber 4.35.

```

public function actionIndex()
{
    $searchModel = new ArTransactionSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

Kode Sumber 4. 35. Melihat Daftar Account Receivable Transaction

4.6.4. Menambah Account Receivable Transaction

Pada implementasi menambah *Account Receivable Transaction*, sistem menambahkan transaksi baru ke dalam *Account Receivable Transaction*. Pengaturan manambah transaksi baru ini diatur pada kelas *ArTransactionController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada Kode Sumber 4.36.

```

public function actionCreate()
{
    $model = new ArTransaction();
    //$model->date = date('Y-m-d');
    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        //if($model['ar_transaction_id'] != 0){
        = SdSale::findOne($model['socode_sale']);
        $soId = $model['sd_sale_id'];
        $status = $model['ar_transaction_status_id'];
        SdSaleController::actionArStatus($soId, "1")
        SdSaleController::actionPaymentStatus($soId, $status);
        $newId =
        str_pad($model['id'],11,"ARTRANS0000",STR_PAD_LEFT);
        $data = ArTransaction::findOne(['id' =>
        $model['id']]);

        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ])
    }
}

```

Kode Sumber 4. 36. Menambah Account Receivable Transaction

Fungsi *actionCreate()* pada kode sumber 4.36. berfungsi untuk menambahkan transaksi baru dimana ketika transaksi berhasil ditambahkan maka akan mengubah status *Sales Order* yang dipilih menjadi 1. Status tersebut menjadi penanda agar ketika akan menambah transaksi baru, *Sales Order* yang sudah berstatus 1 tidak muncul kembali saat mengisi form menambah transaksi. Penamaan nomor transaksi dibuat otomatis dengan format 'ARTRANS0000', 4 digit terakhir menyesuaikan dengan id transaksi. Status transaksi otomatis berstatus 2, artinya transaksi tersebut belum lunas atau '*incomplete*'. Status transaksi tersebut akan berubah menjadi 1 atau '*complete*' ketika pembayaran sudah dilakukan dan lunas.

4.6.5. Menyunting *Account Receivable Transaction*

Pada implementasi menyunting *Account Receivable Transaction*, sistem memperbarui data transaksi pada *Account Receivable Transaction*. Pengaturan menyunting transaksi ini diatur pada kelas *ArTransactionController* fungsi *actionUpdate()*. Kode untuk menyunting *Account Receivable Transaction* ditunjukkan pada Kode Sumber 4.37.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
        $soId = $model['sd_sale_id'];
        $status = $model['ar_transaction_status_id'];

        SdSaleController::actionPaymentStatus($soId,
        $status);
        return $this->redirect(['view', 'id' => $model-
        >id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}
```

Kode Sumber 4. 37.Menyunting *Account Receivable Transaction*

Fungsi `actionUpdate()` pada kode sumber 4.35. berfungsi untuk memperbarui transaksi dimana ketika transaksi berhasil diperbarui maka juga akan mengubah nilai *amount* pada *Account Receivable Payment* apabila terjadi perubahan nilai total pada *Account Receivable Transaction*.

4.6.6. Menghapus *Account Receivable Transaction*

Pada implementasi menghapus *Account Receivable Transaction*, sistem menghapus data transaksi pada *Account Receivable Transaction*. Pengaturan menyunting transaksi ini diatur pada kelas *ArTransactionController* fungsi `actionDelete()`. Kode untuk menghapus *Account Receivable Transaction* ditunjukkan pada Kode Sumber 4.38.

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();
    return $this->redirect(['index']);
}
```

Kode Sumber 4. 38. Menghapus *Account Receivable Transaction*

4.6.7. Melihat Daftar *Account Receivable Payment*

Pada implementasi melihat daftar *Account Receivable Payment*, sistem menampilkan seluruh daftar pembayaran yang berhasil ditambahkan pada *Account Receivable Payment*. Pengaturan tampilan ini diatur pada kelas *ArPaymentController* fungsi `actionIndex()` Kode untuk menampilkan seluruh pembayaran tersebut ditunjukkan pada Kode Sumber 4.39.

```
public function actionIndex()
{
    $searchModel = new ArPaymentSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
    return $this->render('index', ['searchModel' => $searchModel, 'dataProvider' => $dataProvider,
    ]);
}
```

Kode Sumber 4. 39. Melihat Daftar *Account Receivable Payment*

4.6.8. Menambah *Account Receivable Payment*

Pada implementasi menambah *Account Receivable Payment*, sistem menambahkan pembayaran baru ke dalam *Account Receivable Payment*. Pengaturan menambah pembayaran baru ini diatur pada kelas *ArPaymentController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh pembayaran tersebut ditunjukkan pada Kode Sumber 4.40.

```
public function actionCreate()
{
    $model = new ArPayment();
    // $model->date = date('Y-m-d');
    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll() /* && $armodel->saveAll() */) {
        $newId
    str_pad($model['id'],13,"ARPAYMENT0000",STR_PAD_LEFT);
        $code = ArPayment::findOne(['id' =>
    $model['id']]);
        $code->paymentNumber = $newId;
        $code->update();
        if($model['ar_transaction_id'] != 0){
            $armodel['ar_transaction_status_id'] = '2';
            JournalHeaderController::actionShip('ar', $ref[$key],
            $model['status'], $sold); }
        return $this->redirect(['view', 'id' => $model-
    >id]);
    } else {return $this->render('create', [
        'model' => $model,
    ]);
    }
}
```

Kode Sumber 4. 40. Menambah *Account Receivable Payment*

Fungsi *actionCreate()* pada kode sumber 4.40. berfungsi untuk menambahkan pembayaran baru dimana ketika pembayaran berhasil ditambahkan maka akan mengubah status *Account Receivable Transaction* yang dipilih menjadi 1. Status tersebut menjadi penanda agar ketika akan menambah pembayaran baru, *Account Receivable Transaction* yang sudah berstatus 1 tidak muncul kembali saat mengisi form menambah pembayaran. Nomor transaksi dibuat otomatis dengan format 'ARPAYMENT0000, 4 digit terakhir menyesuaikan dengan id pembayaran. Nominal

pembayaran ditambahkan pada *tabular form ArPaymentLine*. Setiap nominal pembayaran akan masuk ke dalam *Journal Header*. Status pembayaran otomatis '*incomplete*', status tersebut akan berubah menjadi '*complete*' apabila jumlah pembayaran sama dengan jumlah hutang. Status pembayaran ini terhubung dengan *Sales Order* yang ada pada *Account Receivable Transaction* yang dipilih, sehingga *sales staff* juga dapat mengetahui status pembayaran dari masing-masing *Sales Order* yang sudah mereka buat sebelumnya.

4.6.9. Menyunting *Account Receivable Payment*

Pada implementasi menyunting *Account Receivable Payment*, sistem memperbarui data pembayaran pada *Account Receivable Payment*. Pengaturan menyunting transaksi ini diatur pada kelas *ArPaymentController* fungsi *actionUpdate()*. Kode untuk menyunting *account receivable payment* ditunjukkan pada Kode Sumber 4.41.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post())) {
        if($model['ar_transaction_id'] != 0){
            $armodel = ArTransaction::findOne($model['ar_transaction_id']);
        }
    }
}
```

Kode Sumber 4. 41. Menyunting *Account Receivable Payment*

Fungsi *actionUpdate()* pada Kode Sumber 4.41. berfungsi untuk memperbarui pembayaran dimana ketika pembayaran berhasil diperbarui maka juga akan mengubah status pada *Sales Order* apabila terjadi perubahan status pada *Account Receivable Payment*. Fungsi ini juga akan memperbarui *Journal Header* apabila terjadi penambahan nominal pembayaran pada *ArPaymentLine*.

4.6.10. Menghapus *Account Receivable Payment*

Pada implementasi menghapus *Account Receivable Payment*, sistem menghapus data pembayaran pada *Account Receivable Payment*. Pengaturan menghapus pembayaran ini diatur pada kelas *ArPaymentController* fungsi *actionDelete()*. Kode untuk menghapus *account receivable payment* ditunjukkan pada Kode Sumber 4.42.

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}
```

Kode Sumber 4. 42. Menghapus *Account Receivable Payment*

4.6.11. Melihat *Account Receivable Aging Analysis*

Pada implementasi melihat *Account Receivable Aging Analysis*, sistem menampilkan data hutang pelanggan yang telah melewati tanggal jatuh tempo. Pengaturan melihat umur hutang ini diatur pada kelas *ArAgingController* fungsi *actionIndex()*. Kode untuk menampilkan daftar umur hutang pelanggan ditunjukkan pada Kode Sumber 4.43.

```
public function actionIndex()
{
    $query = "SELECT    currency.codeCurrency,
sd_customer.name_customer , ar_transaction.transactionNumber,
ar_payment.residu as amount, ar_transaction.dueDate FROM
currency, ar_transaction, ar_payment, sd_customer WHERE
currency.id          = ar_transaction.currency_id      and
ar_payment.ar_transaction_id = ar_transaction.id      and
sd_customer.id       = ar_transaction.sd_customer_id  and
ar_transaction.ar_transaction_status_id = '2' ORDER BY
sd_customer.name_customer";
    ];
}
```

Kode Sumber 4. 43. Melihat *Account Receivable Aging Analysis*

4.6.12. Melihat *Account Receivable Customer Analysis*

Pada implementasi melihat *Account Receivable Customer Analysis*, sistem menampilkan data pelanggan yang pernah melakukan transaksi pembelian. Pengaturan melihat daftar pelanggan ini diatur pada kelas *ArCustomerController* fungsi *actionIndex()*. Kode untuk menampilkan daftar pelanggan ditunjukkan pada Kode Sumber 4.44.

```
public function actionIndex()
{
    $customergroup          =          Yii::$app->request-
    >post('customergroup');

    $customer = Yii::$app->request->post('customer');

    if($customergroup != ""){
        $title['customergroup'] = $customergroup;
        //get list customer berdasarkan group
        $db = Yii::$app->db;
        $query5 = 'SELECT      id,name_customer      FROM
sd_customer WHERE sd_customergroup_id = ' . $customergroup;
        $command5 = $db->createCommand($query5);
        $results5 = $command5->queryAll();
        $data['dataProvider5'] = $results5;
        //get nama customergroup
        $query8

    =      "SELECT      sd_customergroup.name_customergroup      from
sd_customergroup where sd_customergroup.id = $customergroup";

        $db = Yii::$app->db;
        $command8 = $db->createCommand($query8);
        $results8 = $command8->queryAll();
        $data['dataProvider8'] = $results8;
    ...
        //get list customergroup
        $query7 = 'SELECT      id,name_customergroup      FROM
sd_customergroup';

        return $this->render('index', $data);
    }
}
```

Kode Sumber 4. 44. Melihat Account Receivable Customer Analysis

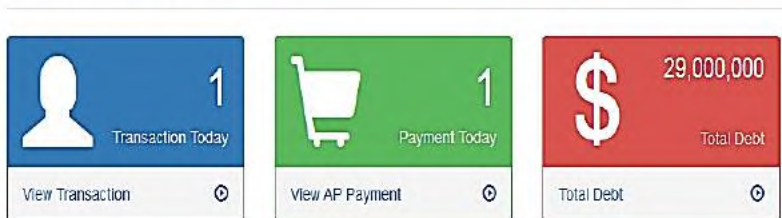
4.7. Implementasi Antarmuka Pengguna *Account Payable*

Pada bagian ini akan dijelaskan mengenai implementasi antarmuka yang terdapat pada Modul *Account Payable* yang terbagi menjadi tampilan halaman utama atau *dashboard* dan beberapa sub modul, antara lain: *Account Payable Transaction*, *Account Payable Payment*, *Account Payable Aging Analysis*, *Account Payable Debt Analysis*. Secara rinci mengenai implementasi lapisan antarmuka Modul *Account Payable* dijabarkan sebagai berikut:

4.7.1. Antarmuka Halaman Utama *Account Payable*

Pada antarmuka ini pengguna dapat melihat ringkasan dari sub-modul yang terdapat pada *Account Payable* . Tampilan antarmuka ini dapat dilihat pada Gambar 4.1.

Account Payable







Gambar 4. 1. Tampilan Antarmuka Halaman Utama *Account Payable*

Pada antarmuka halaman utama, terdapat ringkasan transaksi hari ini (*Transaction Today*), ringkasan pembayaran yang masuk pada hari ini (*Payment Today*) dan total hutang perusahaan yang belum terbayar (*Total Debt*).

4.7.2. Antarmuka Melihat Daftar *Account Payable Transaction*

Pada antarmuka ini pengguna dapat melihat daftar *Account Payable Transaction* yang terdapat pada sistem. Pengguna

juga dapat memilih melakukan kegiatan menyunting, melihat rincian, dan menghapus *Account Payable Transaction* tertentu serta membuat *Account Payable Transaction* baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.2.

#	Po Header	Transaction Number	Date	Due Date	Time Left	Ap Transaction Status ID	Ap Transaction Type
		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Status <input type="text"/>	
1	PO-REQ-Asset-19-1	APTRANS0001	2016-07-11	2016-07-31	13 days	 Complete	From PO
2	PO-REQ-PRO-14-1	APTRANS0002	2016-07-12	2016-08-12	25 days	 Complete	From PO
3	PO-REQ0000013-1	APTRANS0003	2016-06-01	2016-06-10	 Expired	 Incomplete	From PO

Gambar 4. 2. Tampilan Antarmuka Melihat Daftar *Account Payable Transaction*

Pada tampilan antarmuka melihat daftar *account payable transaction* terdapat kolom *Po Header* yang berisi nomor *purchase order*(PO), kolom *Transaction Number* yang berisi nomor transaksi, kolom *Date* yang berisi tanggal pembuatan transaksi, kolom *Due Date* yang berisi tanggal jatuh tempo hutang, kolom *Time Left* yang berisi sisa waktu pembayaran dari hari ini sampai tanggal jatuh tempo. Kolom *AP Transaction Status* merupakan status transaksi yang terdiri dari *Complete* (lunas) dan *Incomplete* (belum lunas). Kolom *Ap Transaction Type* merupakan tipe transaksi yang dibuat. Terdapat 2 tipe transaksi yaitu *From PO* (yang berasal dari *purchase order*) dan *New Transaction*, yaitu transaksi baru tanpa melalui *purchase order*.

4.7.3. Antarmuka Menambah *Account Payable Transaction*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Account Payable Transaction* dan menekan tombol “*Create*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.3.

The screenshot shows a web application interface for adding a payable transaction. It features several dropdown menus for selecting transaction type, purchase order header, supplier, and currency. Below these are three tabs: 'Basic', 'Detail', and 'Dunning'. The 'Basic' tab is currently selected, displaying input fields for transaction name, amount, total, date, due date, and a note. At the bottom of the form are two buttons: 'Create' (green) and 'Cancel' (red).

Gambar 4. 3. Tampilan Antarmuka Menambah *Account Payable Transaction*

Pada Antarmuka menambah *Account Payable Transaction* terdapat *dropdown Ap Transaction Type ID* dimana pengguna dapat memilih tipe transaksi. Terdapat 2 tipe transaksi yaitu *From PO* apabila transaksi berasal dari PO/*Purchase Order*, dan *New Transaction* apabila pembuatan transaksi tanpa melalui *Purchase Order*. *Dropdown Pur Po Header ID* berisi daftar nomor PO, *dropdown Supplier ID* berisi nama *supplier*, *Currency ID* merupakan mata uang yang digunakan. Untuk pilihan pada *Supplier ID* dan *Currency ID* otomatis terisi berdasarkan nomor PO yang dipilih.

Pada tab *Basic* pengguna mengisi *textbox Transaction Name* yaitu nama transaksi, *datepicker Date* yaitu tanggal pembuatan transaksi, *datepicker Due Date* yaitu tanggal jatuh tempo. Dan *textbox amount* yaitu nilai tagihan pada *purchase order* berdasarkan nomor PO yang dipilih.

4.7.4. Antarmuka Menyunting *Account Payable Transaction*

Pada antarmuka ini pengguna dapat merubah data *Account Payable Transaction* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.4.

Ap Transaction Type ID
From PO

Pur Po Header ID: PO-REQ0000001-1
Supplier ID: REBN JAYA
Currency ID: IDR

Basic Detail Dunning

Transaction Name: TRANS1
Amount: 5637500
Total: 5535000

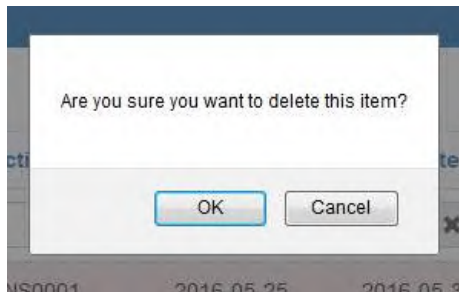
Date: 2016-05-21
Due Date: 2016-05-31
Note:

Update Cancel

Gambar 4. 4. Tampilan Antarmuka Menyunting *Account Payable Transaction*

4.7.5. Antarmuka Menghapus *Account Payable Transaction*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “OK” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.5.



Gambar 4. 5. Antarmuka Menghapus *Account Payable Transaction*

4.7.6. Antarmuka Melihat Daftar *Account Payable Payment*

Pada antarmuka ini pengguna dapat melihat daftar *Account Payable Payment* yang terdapat pada sistem. Pengguna

juga dapat memilih melakukan kegiatan menyunting, melihat rincian, dan menghapus *Account Payable Payment* tertentu serta membuat *Account Payable Payment* baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.6.

#	Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status	Note
		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Status: <input type="text"/>	<input type="text"/>
1	APTRANS0001	APPAYMENT0001	2016-07-31	13 days	IDR 2,200,000,000	IDR 2,200,000,000	IDR 0	<input type="button" value="Complete"/>	
2	APTRANS0002	APPAYMENT0002	2016-08-12	25 days	IDR 263,756,250	IDR 263,756,250	IDR 0	<input type="button" value="Complete"/>	

Gambar 4. 6. Tampilan Antarmuka Melihat Daftar *Account Payable Payment*

Pada tampilan antarmuka melihat daftar *account payable payment* terdapat kolom *Transaction Number* yang berisi nomor transaksi, kolom *Payment Number* yang berisi nomor pembayaran, kolom *Date* yang berisi tanggal pembayaran, kolom *Due Date* yang berisi tanggal jatuh tempo, kolom *Time Left* yang berisi sisa waktu pembayaran dari hari ini sampai tanggal jatuh tempo, kolom *Amount* yang berisi total hutang, kolom *Paid* berisi hutang yang telah terbayar dan kolom *Residu* yang berisi sisa hutang yang belum terbayar. Kolom Status merupakan status pembayaran yang terdiri dari *Complete* (lunas) dan *Incomplete* (belum lunas). Kemudian terdapat kolom *Note* yang berisi catatan tambahan pada transaksi.

4.7.7. Antarmuka Menambah *Account Payable Payment*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Account Payable Payment* dan menekan tombol “*Create*” untuk perintah pembuatan pembayaran . Tampilan antarmuka ini dapat dilihat pada Gambar 4.7.

Gambar 4. 7. Tampilan Antarmuka Menambah *Account Payable Payment*

Pada Antarmuka menambah *Account Payable Payment* terdapat *dropdown Ap Transaction ID* yang berisi daftar nomor transaksi, *dropdown Supplier ID* berisi nama *supplier*, *Currency ID* berisi daftar mata uang. Pada *dropdown Supplier ID* dan *Currency ID* otomatis terisi berdasarkan nomor transaksi yang dipilih. Pengguna dapat mengisi nama pembayaran pada *textbox Payment Name*, tanggal pembuatan transaksi pada *datepicker Date*, tanggal jatuh tempo pada *datepicker Due Date*. Nilai pada *textbox Amount* berasal dari nomor transaksi yang dipilih. *Textbox Paid* berisi jumlah utang yang telah terbayarkan, sedangkan *textbox Residu* berisi sisa hutang yang belum terbayarkan.

Pada *form AP Payment Line* pengguna dapat mengisi jumlah pembayaran yang akan ditambahkan yang terdiri dari *datepicker DatePay* yaitu tanggal pembayaran, *textbox Payment Line Number* yaitu nomor pembayaran, *textbox Payment Line Name* yaitu nama pembayaran, *textbox Pay* yaitu nominal uang yang dibayarkan. *Dropdown GL Journal* yaitu daftar jurnal yang akan dipilih. Pada pembayaran hutang, jurnal yang dipilih adalah jurnal Pembelian Barang Mentah. Kemudian terdapat *textbox Note* untuk menambahkan catatan pembayaran.

4.7.8. Antarmuka Menyunting *Account Payable Payment*

Pada antarmuka ini pengguna dapat merubah data *Account Payable Payment* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.8.

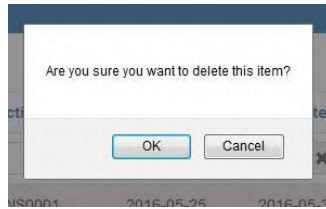
Ap Transaction ID APTRANS0001	Currency ID IDR	Supplier ID REBIN JAYA
Payment Name PAYMENT 1	Amount 5535000	
Date 2016-05-21	Paid 5535000	
Due Date 2016-05-31	Residu 0	
Date Completed 2016-05-23	Status COMPLETE	
Note Note		

Ap Payment Line						
#	Date Pay	Payment Line Number	Payment Line Name	Pay	GI Journal	
1	20	PAY0001	PAYMENT 1	2000000	Pembelian	
2	20	PAY0003	PAYMENT 3	2000000	Pembelian	
3	20	PAY0004	PAYMENT 4	1535000	Pembelian	

Gambar 4. 8. Antarmuka Menyunting *Account Payable Payment*

4.7.9. Antarmuka Menghapus *Account Payable Payment*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “*Yes*” atau pembatalan dengan menekan tombol “*Cancel*”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.9.



Gambar 4. 9. Tampilan Antarmuka Menghapus *Account Payable Payment*

4.7.10. Antarmuka Melihat *Account Payable Aging Analysis*

Pada antarmuka ini pengguna dapat melihat daftar hutang perusahaan berdasarkan tanggal lewat jatuh tempo. Tampilan antarmuka ini dapat dilihat pada Gambar 4.10.

AP Aging Analysis: 18 Jul 2016								
No	Transaction Number	Amount	Due Date	Current	1-30 Days	31-60 Days	61-90 Days	>90 Days
Wim Supply								
1	APTRAVS0003	IDR 29,000,000.00	2016-06-10	-	-	IDR 29,000,000.00	-	-
Sub Total		IDR 29,000,000.00	-			IDR 29,000,000.00		
Total		IDR 29,000,000.00	-			IDR 29,000,000.00		

Gambar 4. 10. Tampilan Antarmuka Melihat *Account Payable Aging Analysis*

Tampilan antarmuka *Account Payable Aging Analysis* dibuat berdasarkan tabel *Aging Analysis* sesuai referensi [15] , terdapat kolom *Transaction Number* yaitu nomor transaksi, *Amount* yaitu jumlah hutang pada satu transaksi, *Due Date* yaitu tanggal jatuh tempo, kolom *Current* terisi apabila transaksi tersebut belum melewati tanggal jatuh tempo, sedangkan kolom *1-30 Days*, *31-60 Days*, *61-90 Days* akan terisi apabila transaksi telah melewati tanggal jatuh tempo sesuai total hari keterlambatan. Kemudian total hutang dari masing masing *supplier* dapat dilihat

pada baris Sub Total. Sedangkan jumlah hutang pada semua *supplier* berada pada kolom Total.

4.7.11. Antarmuka Melihat *Account Payable Debt Analysis*

Pada antarmuka ini pengguna dapat melihat daftar ranking hutang berdasarkan tingkat kepentingannya. Tampilan antarmuka ini dapat dilihat pada Gambar 4.11.

Debt Analysis									
Transaction Number	Transaction Name	Due Date	Fine Per Month(IDR)	Total Debt(IDR)	Payment Tolerance(Days)	Time Left (Days)	Late Frequency(%)	Discount Early Payment(IDR)	Scores
APTRANS0003	Transaction PO 13	2016-06-10	1450000	IDR 29.000.000.00	7	8	0	IDR 600.000.00	0.148
APTRANS0002	Transaction PRO 14	2016-08-12	0	IDR 0.00	7	25	0	IDR 4.613.750.00	0.295
APTRANS0001	Transaksi 1	2016-07-31	0	IDR 0.00	7	13	0	IDR 40.000.000.00	0.373

Gambar 4. 11. Tampilan Antarmuka Melihat *Account Payable Debt Analysis*

Pada antarmuka AP Debt Analysis terdapat kolom *Transaction Number* yang berisi nomor transaksi, *Transaction Name* berisi Nama Transaksi, *Due Date* berisi Tanggal jatuh tempo, *Fine Per Month* yaitu nilai pada kriteria denda yang didapatkan apabila membayar hutang melewati jatuh tempo, *Total Debt* berisi jumlah hutang yang belum terbayar, *Payment Tolerance* berisi toleransi waktu pembayaran yang diberikan supplier, *Time Left* berisi sisa waktu mendekati jatuh tempo, *Late Frequency* berisi frekuensi keterlambatan pembayaran yang dilakukan oleh perusahaan pada *supplier* tertentu, *Discount Early Payment* berisi diskon yang didapat apabila melakukan pelunasan lebih awal. Dari masing masing kriteria tersebut dilakukan pembobotan dengan metode AHP kemudian bobot dari masing-masing kriteria dijumlahkan dalam kolom skor. Data ditampilkan sesuai urutan skor terendah.

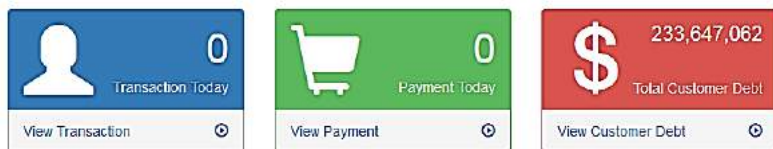
4.8. Implementasi Antarmuka Pengguna *Account Receivable*

Pada bagian ini akan dijelaskan mengenai implementasi antarmuka yang terdapat pada Modul *Account Receivable* yang terbagi menjadi tampilan halaman utama atau *dashboard* dan beberapa sub modul, antara lain: *Account Receivable Transaction*, *Account Receivable Payment*, *Account Receivable Aging Analysis*, *Account Receivable Customer Analysis*, *Account Receivable Report*. Secara rinci mengenai implementasi lapisan antarmuka Modul *Account Receivable* dijabarkan sebagai berikut:

4.8.1. Antarmuka Halaman Utama *Account Receivable*

Pada antarmuka ini pengguna dapat melihat ringkasan dari sub-modul yang terdapat pada *Account Receivable*. Tampilan antarmuka ini dapat dilihat pada Gambar 4.12.

Account Receivable



Gambar 4. 12. Tampilan Antarmuka Halaman Utama *Account Receivable*

Pada antarmuka halaman utama, terdapat ringkasan transaksi hari ini (*Transaction Today*), ringkasan pembayaran yang masuk pada hari ini (*Payment Today*) dan total piutang perusahaan yang belum terbayar (*Total Debt*).

4.8.2. Antarmuka Melihat Daftar *Account Receivable Transaction*

Pada antarmuka ini pengguna dapat melihat daftar *Account Receivable Transaction* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan menyunting,

melihat rincian, dan menghapus *Account Receivable Transaction* tertentu serta membuat *Account Receivable Transaction* baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.13.

#	Transaction Number	Date	Due Date	Time Left	Ar Transaction Status ID	Note
	<input type="text"/>	<input type="text"/>	<input type="text"/>		Status	
1	ARTRANS0011	2016-07-24	2016-07-05	Expired	Bad Debt	
2	ARTRANS0012	2016-07-12	2016-07-31	13 days	Complete	

Gambar 4. 13. Tampilan Antarmuka Melihat Daftar *Account Receivable Transaction*

Pada tampilan antarmuka melihat daftar *account receivable transaction* terdapat kolom *SD Sale ID* yang berisi nomor *sales order(SO)*, kolom *Transaction Number* yang berisi nomor transaksi, kolom *Date* yang berisi tanggal pembuatan transaksi, kolom *Due Date* yang berisi tanggal jatuh tempo hutang, kolom *Time Left* yang berisi sisa waktu pembayaran dari hari ini sampai tanggal jatuh tempo. Kolom *AR Transaction Status* merupakan status transaksi yang terdiri dari *Complete* (lunas) dan *Incomplete* (belum lunas). Kolom *Ap Transaction Type* merupakan tipe transaksi yang dibuat. Terdapat 2 tipe transaksi yaitu From SO (yang berasal dari *sales order*) dan New Transaction, yaitu transaksi baru tanpa melalui *sales order*.

4.8.3. Antarmuka Menambah *Account Receivable Transaction*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Account Receivable Transaction* dan menekan tombol “*Create*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.14.

Ar Transaction Type ID
Choose Ar transaction type

Sd Sale ID
Choose Sd sale

Sd Customer ID
Choose Customer

Ar Payterm ID
Choose Ar payterm

Currency ID
Choose Currenc

Basic Detail Dunning

Transaction Name
TransactionName

Amount
Amount

Total
Total

Date
Choose Date

Due Date
Choose DueDate

Note
Note

Create Cancel

Gambar 4. 14. Tampilan Antarmuka Menambah *Account Receivable Transaction*

Pada tampilan antarmuka melihat daftar *account receivable transaction* terdapat kolom *Sd Sale ID* yang berisi nomor *Sales Order(SO)*, kolom *Transaction Number* yang berisi nomor transaksi, kolom *Date* yang berisi tanggal pembuatan transaksi, kolom *Due Date* yang berisi tanggal jatuh tempo hutang, kolom *Time Left* yang berisi sisa waktu pembayaran dari hari ini sampai tanggal jatuh tempo. Kolom *AR Transaction Status* merupakan status transaksi yang terdiri dari *Complete* (lunas) dan *Incomplete* (belum lunas). Kolom *Ar Transaction Type* merupakan tipe transaksi yang dibuat. Terdapat 2 tipe transaksi yaitu *From SO* (yang berasal dari *sales order*) dan *New Transaction*, yaitu transaksi baru tanpa melalui *sales order*.

4.8.4. Antarmuka Menyunting *Account Receivable Transaction*

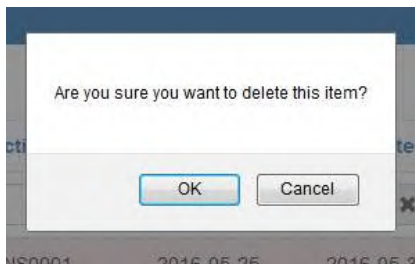
Pada antarmuka ini pengguna dapat merubah data *Account Receivable Transaction* dan menekan tombol “Save” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.15.

Ar Transaction Type ID		
Prepaid		
Sd Sale ID	Sd Customer ID	Ar Payterm ID
24	Boy	Payment discount and fine for 10 days
		Currency ID
		Doll
<input type="button" value="Basic"/> <input type="button" value="Detail"/> <input type="button" value="Listing"/>		
Transaction Name	Amount	Total
Receive 22	304022500.0000	304022500
Date	Due Date	Note
2016-05-23	2016-05-23	Note
<input type="button" value="Update"/> <input type="button" value="Cancel"/>		

Gambar 4. 15. Tampilan Antarmuka Menyunting *Account Receivable Transaction*

4.8.5. Antarmuka Menghapus *Account Receivable Transaction*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “Yes” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.16.



Gambar 4. 16. Tampilan Antarmuka Menghapus *Account Receivable Transaction*

4.8.6. Antarmuka Melihat Daftar *Account Receivable Payment*

Pada antarmuka ini pengguna dapat melihat daftar *Account Payable Transaction* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan menyunting, melihat rincian, dan menghapus *Account Payable Transaction* tertentu serta membuat *Account Payable Transaction* baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.17.

#	Transaction Number	Payment Number	Date	Due Date	Time Left	Amount	Paid	Residu	Status
		<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	Status: ▾
1	ARTRANS0040	ARPAYMENT0001	2016-05-29	2016-06-02	3 days	2213905862	2213905862	0	A Complete
2	ARTRANS0036	ARPAYMENT0002	2016-05-26	2016-05-28	X Incomplete	2594445923	2594445923	0	A Complete
3	ARTRANS0044	ARPAYMENT0003	2016-05-29	2016-05-04	5 days	948815798	940000000	8815798	X Incomplete

Gambar 4. 17. Tampilan Antarmuka Melihat Daftar Account Receivable Payment

Pada tampilan antarmuka melihat daftar *account receivable payment* terdapat kolom *Transaction Number* yang berisi nomor transaksi, kolom *Payment Number* yang berisi nomor pembayaran, kolom *Date* yang berisi tanggal pembayaran, kolom *Due Date* yang berisi tanggal jatuh tempo, kolom *Time Left* yang berisi sisa waktu pembayaran dari hari ini sampai tanggal jatuh tempo, kolom *Amount* yang berisi total hutang, kolom *Paid* berisi hutang yang telah terbayar dan kolom *Residu* yang berisi sisa hutang yang belum terbayar. Kolom Status merupakan status pembayaran yang terdiri dari *Complete* (lunas) dan *Incomplete* (belum lunas). Kemudian terdapat baris yang berwarna merah, kuning, hijau yang menandakan sisa waktu pembayaran dari transaksi tersebut. Apabila sisa waktu jatuh tempo kurang dari 3 hari maka berwarna merah, apabila kurang dari 5 hari akan berwarna kuning dan apabila kurang dari 7 hari akan berwarna hijau. Sedangkan apabila lebih dari 7 hari maka baris berwarna putih.

4.8.7. Antarmuka Menambah Account Receivable Payment

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Account Receivable Transaction* dan menekan tombol “*Create*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.18.

Ar Transaction ID Choose Ar transaction	Ar Payterm ID Choose Ar payterm	Currency ID Choose Currency	Customer ID Choose Customer
Payment Name PaymentName		Amount Amount	
Date Choose Date		Paid Paid	
Due Date Choose DueDate		Residu Residu	
Date Completed Choose DateCompleted		Status Status	
Note Note			

Gambar 4. 18. Tampilan Antarmuka Menambah *Account Receivable Payment*

Pada Antarmuka menambah *Account Receivable Payment* terdapat *dropdown Ar Transaction ID* yang berisi daftar nomor transaksi, *dropdown Customer ID* berisi nama *customer*, *Currency ID* berisi daftar mata uang. Pada *dropdown Customer ID* dan *Currency ID* otomatis terisi berdasarkan nomor transaksi yang dipilih. Pengguna dapat mengisi nama pembayaran pada *textbox Payment Name*, tanggal pembuatan transaksi pada *datepicker Date*, tanggal jatuh tempo pada *datepicker Due Date*. Nilai pada *textbox Amount* berasal dari nomor transaksi yang dipilih. *Textbox Paid* berisi jumlah utang yang telah terbayarkan, sedangkan *textbox Residu* berisi sisa hutang yang belum terbayarkan.

Pada *form AR Payment Line* pengguna dapat mengisi jumlah pembayaran yang akan ditambahkan yang terdiri dari *datepicker DatePay* yaitu tanggal pembayaran, *textbox Payment Line Number* yaitu nomor pembayaran, *textbox Payment Line Name* yaitu nama pembayaran, *textbox Pay* yaitu nominal uang yang dibayarkan. *Dropdown GL Journal* yaitu daftar jurnal yang akan dipilih. Pada pembayaran hutang, jurnal yang dipilih adalah jurnal Penjualan. Kemudian terdapat *textbox Note* untuk menambahkan catatan pembayaran.

4.8.8. Antarmuka Menyunting *Account Receivable Payment*

Pada antarmuka ini pengguna dapat merubah data *Account Receivable Transaction* dan menekan tombol “Save” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.19.

Ar Transaction ID ARTRANS0040 x	Ar Payterm ID Payterm discount and fine for 10 days be x	Currency ID IDR x	Customer ID lan
Payment Name Receive 22		Amount 221390562	
Date 2016-05-29 x		Paid 221390562	
Due Date 2016-06-02 x		Residu 0	
Date Completed 2016-05-29 x		Status COMPLETE	

Gambar 4. 19. Tampilan Antarmuka Menyunting *Account Receivable Payment*

4.8.9. Antarmuka Menghapus *Account Receivable Payment*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “Yes” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.20.

Are you sure you want to delete this item?

OK
Cancel

Gambar 4. 20. Tampilan Antarmuka Menghapus *Account Receivable Payment*

4.8.10. Antarmuka Melakukan Konfirmasi *Sales Order*

Pada antarmuka ini pengguna dapat melihat daftar ranking hutang berdasarkan tingkat kepentingannya. Tampilan antarmuka ini dapat dilihat pada Gambar 4.21.

AR Approve: 12 Jun 2016

Quotation Number	Customer	Amount	Valid Date	Approve
QO00015	Iloal GR	9025939094.7	2016-06-04 00:00:00	

Gambar 4. 21. Antarmuka Melakukan Konfirmasi *Sales Order*

4.8.11. Antarmuka Melihat Daftar *Account Receivable Customer Analysis*

Pada antarmuka ini pengguna dapat melihat urutan customer terbaik berdasarkan data pembelian sebelumnya. Tampilan antarmuka ini dapat dilihat pada Gambar 4.22.

Customer Analysis

Choose Customer Group

Enterprise

Submit

Customer Group: Enterprise

Customer Name	Late Frequency (%)	Bad Debts (IDR)	Paid Before Maturity (%)	Total Payment (IDR)	Payment Complete Average(Month)	Order Frequency	Score
I Gede Arya Putra Perdana	0	0.00	0	1,200,848,000.00		0	1

Gambar 4. 22. Tampilan Antarmuka Melihat Daftar *Account Receivable Customer Analysis*

Pada antarmuka *AP Debt Analysis* terdapat kolom *Transaction Number* yang berisi nomor transaksi, *Transaction Name* berisi nama transaksi, *Due Date* berisi tanggal jatuh tempo, *Fine per Month* yaitu nilai pada kriteria denda yang didapatkan apabila membayar hutang melewati jatuh tempo, *Total Debt* berisi jumlah hutang yang belum terbayar, *Payment Tolerance* berisi toleransi waktu pembayaran yang diberikan *supplier*, *Time Left*

berisi sisa waktu mendekati jatuh tempo, *Late Frequency* berisi frekuensi keterlambatan pembayaran yang dilakukan oleh perusahaan pada *supplier* tertentu, *Discount Early Payment* berisi diskon yang didapat apabila melakukan pelunasan lebih awal. Dari masing masing kriteria tersebut dilakukan pembobotan dengan metode AHP, kemudian bobot dari masing-masing kriteria dijumlahkan dalam kolom Score. Data ditampilkan sesuai urutan skor terbanyak.

4.8.12. Antarmuka Melihat *Account Receivable Aging Analysis*

Pada antarmuka ini pengguna dapat melihat daftar hutang pelanggan berdasarkan tanggal batas akhir pembayaran. Tampilan antarmuka ini dapat dilihat pada Gambar 4.23.

No	Transaction Number	Amount	Due Date	Current	1-30 Days	31-60 Days	61-90 Days	>90 Days
Wim Supply								
1	APTRAN00003	IDR 29,000,000.00	2016-05-10	-	-	IDR 29,000,000.00	-	-
	Sub Total	IDR 29,000,000.00	-			IDR 29,000,000.00		
	Total	IDR 29,000,000.00	-			IDR 29,000,000.00		

Gambar 4. 23. Tampilan Antarmuka Melihat *Account Receivable Aging Analysis*

Tampilan antarmuka *Account Receivable Aging Analysis* dibuat berdasarkan tabel *Aging Analysis* sesuai referensi [15], terdapat kolom *Transaction Number* yaitu nomor transaksi, *Amount*, yaitu total piutang pada satu transaksi, *Due Date* yaitu tanggal jatuh tempo, kolom *Current* terisi apabila transaksi tersebut belum melewati tanggal jatuh tempo, sedangkan kolom *1-30 Days*, *31-60 Days*, *61-90 Days* akan terisi apabila transaksi telah melewati tanggal jatuh tempo sesuai total hari keterlambatan. Kemudian total piutang dari masing masing *customer* dapat dilihat pada baris *Sub Total*. Sedangkan jumlah hutang pada semua *customer* berada pada kolom *Total*.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada program yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsional program. Pengujian fungsionalitas akan mengacu pada kasus penggunaan program yang sudah dibahas pada bab tiga terhadap program yang dikembangkan. Hasil dari pengujian tersebut akan dijabarkan pada bagian evaluasi di akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan uji coba untuk aplikasi ini dilakukan pada sebuah komputer. Berikut ini spesifikasi perangkat lunak dan perangkat keras komputer yang digunakan pada uji coba.

- 1) Perangkat keras
 - a. Prosesor: AMD A6-3400M APU with Radeon HD Graphics @1.40Ghz
 - b. Memori (RAM) : 4GB
 - c. Harddisk 500GB
- 2) Perangkat Lunak
 - a. Sistem Operasi : Windows 7 Ultimate
 - b. Perangkat Pengembang : Python-2.7.11

5.2. Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam 3 (tiga) tahap, yaitu pengujian kebutuhan fungsionalitas, pengujian kegunaan sistem, dan pengujian perbandingan *metrics*. Pengujian kebutuhan fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada hasil keluaran sistem.

Perencanaan Bisnis

- **Daftar Barang Jadi**

Tabel 5.1. Daftar Barang Jadi

No	Nama Barang
1	<i>Red Deluxe Touring Bike</i>
2	<i>Red Profesional Touring Bike</i>

Pada Tabel 5.1, ditunjukkan bahwa perusahaan ini memproduksi dua jenis sepeda. Dua jenis sepeda tersebut memiliki bahan dasar yang berbeda, *Deluxe Touring Bike* berbahan dasar aluminum, sedangkan *Profesional Touring Bike* berbahan dasar karbon.

- **Daftar Aset**

Tabel 5.2. Daftar Aset

No.	Nama	Kuantitas
1	Tanah	180 x 150 m ²
2	Kantor	50 x 50 m ²
3	Parkir	60 x 50 m ²
4	Kantin	40 x 60 m ²
5	Pabrik	108 x 80 m ²
6	Pengolahan Limbah	20 x 40 m ²
7	<i>Raw Materials Inventory</i>	40 x 60 m ²
8	<i>Semi-Finished Good Inventory</i>	40 x 60 m ²
9	<i>Finished Good Inventory</i>	40 x 60 m ²
10	<i>Welding Machine</i>	2 baris
11	<i>Molding Machine</i>	2 baris
12	<i>Laser Cutting Machine</i>	2 baris
13	<i>Spray Painting Machine</i>	2 baris
14	<i>Testing Machine</i>	2 baris
15	Truk	10 buah
16	<i>Forklift</i>	6 buah
17	<i>Heavy Forklift</i>	6 buah

Pada Tabel 5.2, perusahaan memiliki 17 asset pada perencanaannya dengan kuantitas dari masing-masing aset yang ada.

- **Daftar Bahan Baku**

Tabel 5.3. Daftar Bahan Baku

No.	Nama bahan baku	Kuantitas	Satuan unit
1	<i>Tire</i>	1.2	<i>Pcs</i>
2	<i>Seat Kit</i>	0.5	<i>Pcs</i>
3	<i>Chain</i>	1.5	<i>Pcs</i>
4	<i>Gear</i>	1.8	<i>Pcs</i>
5	<i>Brake</i>	0.6	<i>Pcs</i>
6	<i>Handle Bar</i>	1.2	<i>Pcs</i>
7	<i>Pedal</i>	0.4	<i>Pcs</i>
8	<i>Aluminium</i>	2	m
9	<i>Carbon Fiber</i>	2	m
10	<i>Red Paint 20KG</i>	20	<i>Big Drum</i>
11	<i>Velg</i>	2.3	<i>Pcs</i>
12	<i>Tube</i>	0.4	<i>Pcs</i>
13	<i>Hex Nut 5mm</i>	0.04	<i>Pcs</i>
14	<i>Lock Washer 5mm</i>	0.06	<i>Pcs</i>
15	<i>Socket Head Bolt 5mm</i>	0.03	<i>Pcs</i>

Perencanaan yang diadakan oleh perusahaan ini akan menggunakan 15 macam bahan baku seperti yang dijabarkan pada Tabel 5.3. 15 bahan baku ini ditentukan dari daftar material yang akan diterangkan di Tabel 5.4.

- **Daftar Material**

Tabel 5.4. Daftar Material

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan
1	<i>Aluminium</i>	5	m	Kerangka untuk
2	<i>Paint drum with Red Color 20KG</i>	0.05	<i>Big drum</i>	Red Deluxe Touring Bike

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan
3	<i>Carbon Fiber</i>	5	m	Kerangka untuk Red Profesional Touring Bike
4	<i>Paint drum with Red Color 20KG</i>	0.05	<i>Big drum</i>	
5	<i>Hex Nut 5 mm</i>	2	<i>Pcs</i>	
6	<i>Lock Washer 5 mm</i>	2	<i>Pcs</i>	
7	<i>Socket Head Bolt 5mm</i>	1	<i>Pcs</i>	
8	<i>Tire's Bike</i>	1	<i>Pcs</i>	
9	<i>Wheel's Tube</i>	1	<i>Pcs</i>	
10	<i>Wheel's Velg</i>	1	<i>Pcs</i>	
11	<i>Frame for Deluxe Touring Bike with Red Color</i>	1	<i>Pcs</i>	Red Deluxe Touring Bike
12	<i>Wheel's Bike</i>	2	<i>Pcs</i>	
13	<i>Chain's Bike</i>	1	<i>Pcs</i>	
14	<i>Handle Bar's Bike</i>	1	<i>Pcs</i>	
15	<i>Seat Kit's Bike</i>	1	<i>Pcs</i>	
16	<i>Pedal's Bike</i>	2	<i>Pcs</i>	
17	<i>Gear's Bike</i>	1	<i>Pcs</i>	
18	<i>Front and Rear Brake's Bike</i>	1	<i>Pcs</i>	
19	<i>Frame for Deluxe Touring Bike with Red Color</i>	1	<i>Pcs</i>	Red Professional Touring Bike
20	<i>Wheel's Bike</i>	2	<i>Pcs</i>	
21	<i>Chain's Bike</i>	1	<i>Pcs</i>	
22	<i>Handle Bar's Bike</i>	1	<i>Pcs</i>	
23	<i>Seat Kit's Bike</i>	1	<i>Pcs</i>	
24	<i>Pedal's Bike</i>	2	<i>Pcs</i>	
25	<i>Gear's Bike</i>	1	<i>Pcs</i>	
26	<i>Front and Rear Brake's Bike</i>	1	<i>Pcs</i>	

Daftar material yang dijabarkan pada tabel 5.4. menjelaskan material-material yang dibutuhkan untuk memproduksi barang jadi yang dibutuhkan pada perusahaan ini.

- **Sumber Daya**

Pada perusahaan ini memiliki beberapa sumber daya yang terdiri atas beberapa pegawai dan beberapa mesin yang terpisah menjadi 2 (dua) baris. Terdapat 150 orang sebagai *direct labour*, 32 orang sebagai *indirect labour*. Untuk mesin, terdapat *welding machine*, *molding machine*, *laser cutting machine*, *spray machine*, dan *testing machine* yang masing-masing mesin untuk 2 (dua) baris.

- **Perencanaan keuangan**

Perencanaan keuangan dalam perusahaan ini digunakan untuk merincikan keuangan yang digunakan untuk pengelolaan keuangan dalam perusahaan. Perencanaan keuangan ditunjukkan pada Table 5.5.

Tabel 5.5. Perencanaan Keuangan Perusahaan

No.	Nama perencanaan keuangan	Jumlah (dalam Rp)
1	Total pembelian aset aktif	27.997.311.200,00
2	Total pembelian bahan baku	88.615.744.062,50
3	Biaya lain-lain	3,386,944,737.50
4	Peminjaman dari Bank	120.000.000.000,00

5.2.1. Pengujian Basis Data Terdistribusi

Pengujian Basis Data Terdistribusi (BDT) dilakukan untuk menguji 2 aspek utama dalam konsep BDT, yaitu replikasi dan *high availability*.

Sistem BDT menggunakan 4 server fisik. Dengan rincian sebagai berikut:

1. *Server aplikasi*, adalah server yang di dalamnya terdapat *web server* beserta file aplikasi ERP.

2. *Management Node*, adalah *server* yang berfungsi sebagai pusat pengaturan sistem BDT.
3. *Data Node*, adalah 2 buah *server* penyimpan data.

- **Prosedur Simulasi**

Replikasi

1. Memastikan seluruh *server* dan sistem BDT dalam keadaan *online*.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi ERP yang ditunjukkan pada Gambar 5.3.
3. Melakukan pengecekan terhadap hasil penambahan, perubahan, dan penghapusan data pada seluruh *data node* yang ditampilkan pada Gambar 5.4. dan Gambar 5.5.

High Availability

1. Mematikan *server* aplikasi, *management server* dan sistem BDT pada salah satu *node*. Ditampilkan pada Gambar 5.6.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi yang ditunjukkan pada Gambar 5.7.
3. Melakukan pengecekan terhadap hasil pemrosesan data pada *node* yang masih aktif yang ditampilkan pada Gambar 5.8.

Warehouse 3

ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT

Gambar 5. 1. Pengujian fitur Replikasi pada sistem

```

master@master-Aspire-M3970: ~
mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
+-----+
| id | company_id | ws_name          | created_at          | updated_at          |
+-----+
| 3 | 1 | EZERP Testing BDT | NULL                | NULL                |
| 1 | 1 | Maju              | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL                | NULL                |
+-----+
3 rows in set (0.01 sec)

mysql>

```

Gambar 5. 2. Pengujian Fitur Replikasi pada Database Server 1

```

root@node2-Aspire-M3970: /home/node2
mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
+-----+
| id | company_id | ws_name          | created_at          | updated_at          |
+-----+
| 3 | 1 | EZERP Testing BDT | NULL                | NULL                |
| 1 | 1 | Maju              | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL                | NULL                |
+-----+
3 rows in set (0.00 sec)

mysql>

```

Gambar 5. 3. Pengujian Fitur Replikasi pada Databse Server 2

5.2.2. Pengujian RBAC (Role Based Access Control)

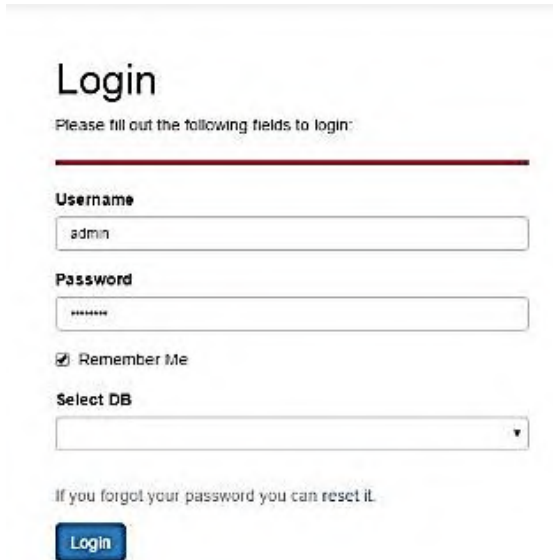
Pengujian ini terdiri dari pengujian mengelola *Role Based Access Control (RBAC)*. Rincian skenario pengujian pada kasus penggunaan pengujian mengelola *Role Based Access Control (RBAC)* dapat dilihat pada Tabel 5.2.

Tabel 5. 1. Pengujian fitur mengelola Role Based Access Control (RBAC)

ID	UJ-UC-RBAC
Referensi Kasus Penggunaan	UC-RBAC
Nama	Pengujian mengelola <i>Role Based Access Control (RBAC)</i> .
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Role Based Access Control (RBAC)</i> .
Skenario 1	<i>Pengguna membuat user baru.</i>
Kondisi Awal	Pengguna berada pada halaman dashboard admin.
Data Uji	Inputan data <i>user</i> baru:
Langkah Pengujian	Pengguna memilih kegiatan <i>Add New User</i> , lalu memasukkan data inputan dan menekan tombol “ <i>Add User</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Yang Didapat	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman halaman daftar <i>user</i> dengan tambahan data <i>user</i> yang baru.
Skenario 2	<i>Pengguna menyunting data user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	Perubahan data <i>user</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>user</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.

Hasil Yang Diharapkan	<i>User yang disunting mengalami perubahan data sesuai inputan.</i>
Hasil Yang Didapat	<i>User yang disunting mengalami perubahan data sesuai inputan.</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar user.
Skenario 3	<i>Pengguna menghapus user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar user.
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “Yes” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>User yang baru dihapus tidak tampil pada daftar user.</i>
Hasil Yang Didapat	<i>User yang baru dihapus tidak tampil pada daftar user.</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar user.

Gambar 5.9 menunjukkan hasil setelah proses login oleh admin. Gambar 5.10 menunjukkan hasil pengujian penginputan data *user* untuk skenario 1, dimana dapat dilihat penambahan pada halaman daftar *user* yang baru. Kemudian pada Gambar 5.11 menunjukkan proses penyuntingan data *user* untuk skenario 2. Sedangkan pada Gambar 5.12 dapat dilihat proses penghapusan *user* untuk skenario 3. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC.RBAC telah bekerja dengan baik seperti yang diharapkan.



Login

Please fill out the following fields to login:

Username

Password

☒ Remember Me

Select DB

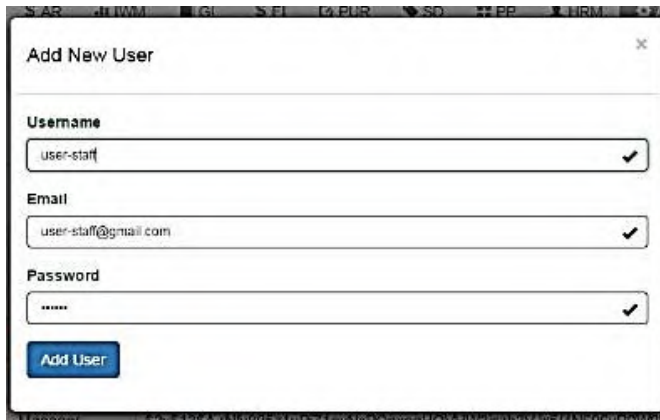
If you forgot your password you can reset it.

Login

Gambar 5. 7. Proses login admin



Gambar 5. 8. Tampilan awal setelah login berhasil dilakukan



Add New User

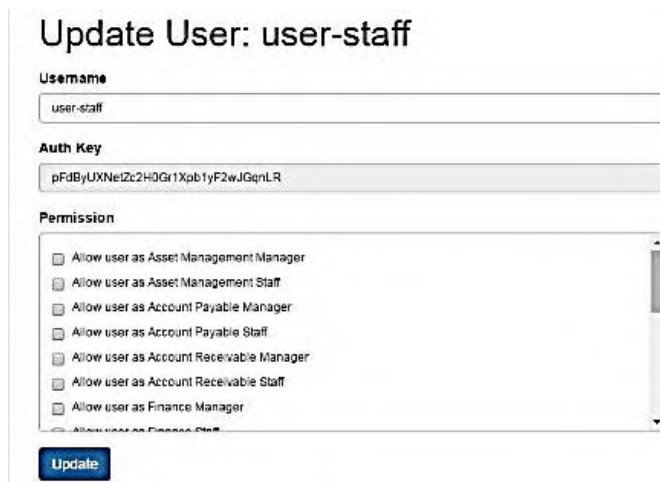
Username
user-staff ✓

Email
user-staff@gmail.com ✓

Password
***** ✓

Add User

Gambar 5. 9. Proses admin menambahkan user baru



Update User: user-staff

Username
user-staff

Auth Key
pFdByUXNetZc2H0Gr1Xpb1yF2wJGqnLR

Permission

- ☐ Allow user as Asset Management Manager
- ☐ Allow user as Asset Management Staff
- ☐ Allow user as Account Payable Manager
- ☐ Allow user as Account Payable Staff
- ☐ Allow user as Account Receivable Manager
- ☐ Allow user as Account Receivable Staff
- ☐ Allow user as Finance Manager

Update

Gambar 5. 10. Proses menyunting data user oleh admin



Gambar 5. 11. Proses penghapusan *user* oleh admin

5.2.3. Pengujian *Multitenancy*

Pengujian ini terdiri dari pengujian mengelola *multitenancy*. Pada pengujian *multitenancy*, *tenant* baru yang akan mendaftarkan *tenant*-nya untuk dapat mengakses sistem memasukkan data berupa nama basis data yang akan dibuat. Apabila basis data berhasil dibuat, maka *tenant* dapat mulai mengakses dengan pengguna awal sebagai administrator. Rincian skenario pengujian pada fitur *multitenancy* dapat dilihat pada Tabel 5.3.

Tabel 5. 2. Pengujian mengelola *multitenancy*

ID	UJ-MTC
Referensi	Multitenancy
Nama	Pengujian mengelola Multitenancy
Tujuan Pengujian	Menguji kemampuan sistem mengelola multitenancy
Skenario 1	<i>Pengguna menambah tenant baru.</i>
Kondisi Awal	Pengguna berada pada halaman utama dan memilih menu tenant
Data Uji	Data Tenant
Langkah Pengujian	Pengguna masuk ke halaman Tenant dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan membuat basis data sesuai dengan nama data uji

Hasil Yang Didapat	Basis data berhasil dibuat
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada <i>login</i>
Skenario 2	Pengguna mengakses Tenant
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Data Uji	Data Tenant
Langkah Pengujian	Pengguna memasukkan data uji ke form yang tersedia dan memilih tombol login
Hasil Yang Diharapkan	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Yang Didapat	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama

5.2.4. Pengujian Fungsionalitas

Pengujian fungsionalitas aplikasi dilakukan secara mandiri dengan melakukan skenario yang sama dengan rancangan alur proses aplikasi sebagai tolok ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

5.2.4.1. Pengujian Fitur Mengelola Transaksi Pembelian

Pengujian ini terdiri dari pengujian membuat, menyunting, dan menyunting *Account Payable Transaction*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.4.

Tabel 5. 3. Pengujian mengelola transaksi pembelian

ID	UJ-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian mengelola <i>Account Payable Transaction</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Account Payable Transaction</i>
Skenario 1	<i>Pengguna membuat Account Payable Transaction baru.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> .
Data Uji	Inputan data <i>Account Payable Transaction</i> :
Langkah Pengujian	Pengguna memilih kegiatan membuat <i>Account Payable Transaction</i> baru, lalu memasukkan data inputan dan menekan tombol “ <i>Create</i> ”.
Hasil Yang Diharapkan	<i>Account Payable Transaction</i> yang baru terdapat pada daftar <i>Account Payable Transaction</i> .
Hasil Yang Didapat	<i>Account Payable Transaction</i> yang baru terdapat pada daftar <i>Account Payable Transaction</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> dengan tambahan data <i>Account Payable Transaction</i> yang baru.
Skenario 2	<i>Pengguna menyunting Account Payable Transaction tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> .
Data Uji	Perubahan data <i>Account Payable Transaction</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>Account Payable Transaction</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>Account Payable Transaction</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>Account Payable Transaction</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> .

Skenario 3	<i>Pengguna menghapus Account Payable Transaction tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> .
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “Yes” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>Account Payable Transaction</i> yang baru dihapus tidak tampil pada daftar <i>Account Payable Transaction</i> .
Hasil Yang Didapat	<i>Account Payable Transaction</i> yang baru dihapus tidak tampil pada daftar <i>Account Payable Transaction</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Transaction</i> .

Gambar 5.14 dan 5.15 menunjukkan proses penginputan data *Account Payable Transaction* untuk skenario 1. Sedangkan untuk Gambar 5.16 menunjukkan hasil pengujiannya, dimana dapat dilihat penambahan *Account Payable Transaction* yang baru. Kemudian pada Gambar 5.17 menunjukkan proses merubah data *Account Payable Transaction*. Hasil pengujiannya dapat dilihat pada Gambar 5.18. Sedangkan pada Gambar 5.19 dapat dilihat proses penghapusan *Account Payable Transaction*. Gambar 5.20 menunjukkan pada daftar bahwa *Account Payable Transaction* telah terhapus. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-001 telah bekerja dengan baik seperti yang diharapkan.

#	Po Header	Transaction Number	Date	Due Date	Time Left	Ap Transaction Status ID	Ap Transaction Type
1	PO-REQ- Forecast-01-1	APTRANS0001	2016-05-31	2016-05-31	Expired	Complete	From PO

Gambar 5. 12. Data awal Account Payable Transaction

Create Ap Transaction

Ap Transaction Type ID
 Choose Ap transaction type ▼

Pur Po Header ID **Supplier ID** **Currency ID**
 Choose PO ▼ Choose Supplier ▼ Choose Currency ▼

Basic **Detail** Dunning

Transaction Name **Amount** **Total**
 TransactionName Amount Total

Date **Due Date** **Note**
 Choose Date Choose DueDate Note

Create Cancel

Gambar 5. 13. Proses penginputan data *Account Payable Transaction*

#	Po Header	Transaction Number	Date	Due Date	Time Left	Ap Transaction Status ID	Ap Transaction Type
						Status ▼	
1	PO-REQ- Forecast-01-1	APTRANS0001	2016-05-25	2016-05-31	X Expired	A Complete	From PO
2	PO-REQ000014-1	APTRANS0002	2016-06-12	2016-06-30	18 days	X Incomplete	From PO

Gambar 5. 14. Data *Account Payable Transaction* yang baru berhasil masuk ke sistem

Update Ap Transaction: 2

Ap Transaction Type ID

From PO

Pur Po Header ID

PO-REQ0000014-1

Supplier ID

Supplier Rantai

Currency ID

IDR

Basic

Detail

Dunning

Transaction Name

Transaction PO 14

Amount

13200000

Total

13200000

Date

2016-06-12

Due Date

2016-06-30

Note

Nota

Update

Cancel

Gambar 5. 15. Proses perubahan data *Account Payable Transaction*

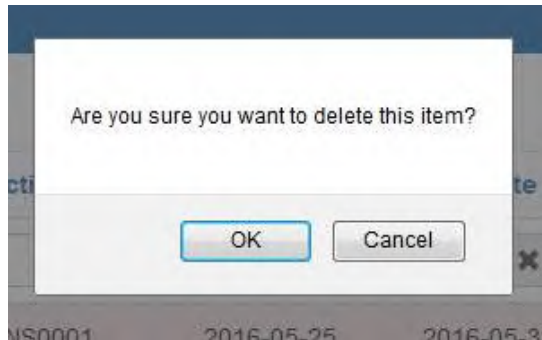
Ap Transaction

Showing 1-2 of 2

Page

#	Po Header	Transaction Number	Date	Due Date	Time Left	Ap Transaction Status ID	Ap Transact
						Status	
1	PO-REQ- Forecast-01-1	APTRANS0001	2016-05-25	2016-05-31	Expired	Complete	From PO
2	PO-REQ0000014-1	APTRANS0002	2016-06-12	2016-06-15	3 days	Incomplete	From PO

Gambar 5. 16. Data *Account Payable Transaction* berhasil diperbarui



Gambar 5. 17. Proses penghapusan *Account Payable Transaction*

Ap Transaction							
Showing 1-1 of 1							
<div> Page Full </div>							
#	Po Header	Transaction Number	Date	Due Date	Time Left	Ap Transaction Status ID	Ap Transaction Type
1	PO-REQ- Forecast-01-1	APTRANS0001	2016-05-26	2016-05-31	Expired	Complete	From PO

Gambar 5. 18. *Account Payable Transaction* berhasil terhapus

5.2.4.2. Pengujian Fitur Mengelola Pembayaran Pembelian

Pengujian ini terdiri dari pengujian membuat, menyunting, dan menyunting *Account Payable Payment*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.5.

Tabel 5. 4. Pengujian mengelola pembayaran pembelian

ID	UJ-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian mengelola <i>Account Payable Payment</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Account Payable Payment</i>
Skenario 1	<i>Pengguna membuat Account Payable Payment baru.</i>

Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> .
Data Uji	Inputan data <i>Account Payable Payment</i> :
Langkah Pengujian	Pengguna memilih kegiatan membuat <i>Account Payable Payment</i> baru, lalu memasukkan data inputan dan menekan tombol “ <i>Create</i> ”.
Hasil Yang Diharapkan	<i>Account Payable Payment</i> yang baru terdapat pada daftar <i>Account Payable Payment</i> .
Hasil Yang Didapat	<i>Account Payable Payment</i> yang baru terdapat pada daftar <i>Account Payable Payment</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> dengan tambahan data <i>Account Payable Payment</i> yang baru.
Skenario 2	<i>Pengguna menyunting Account Payable Payment tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> .
Data Uji	Perubahan data <i>Account Payable Payment</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>Account Payable Payment</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>Account Payable Payment</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>Account Payable Payment</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> .
Skenario 3	<i>Pengguna menghapus Account Payable Payment tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> .
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “ <i>Yes</i> ” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>Account Payable Payment</i> yang baru dihapus tidak tampil pada daftar <i>Account Payable Payment</i> .

Hasil Yang Didapat	<i>Account Payable Payment</i> yang baru dihapus tidak tampil pada daftar <i>Account Payable Payment</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Payable Payment</i> .

Gambar 5.21 dan Gambar 5.22 menunjukkan proses penginputan data *Account Payable Payment* untuk skenario 1. Sedangkan untuk Gambar 5.23 menunjukkan hasil pengujiannya, dimana dapat dilihat penambahan *Account Payable Payment* yang baru. Kemudian pada Gambar 5.24 menunjukkan proses merubah data *Account Payable Payment*. Hasil pengujiannya dapat dilihat pada Gambar 5.25. Sedangkan pada Gambar 5.26 dapat dilihat proses penghapusan *Account Payable Payment*. Gambar 5.27 menunjukkan pada daftar bahwa *Account Payable Payment* telah terhapus. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-002 telah bekerja dengan baik seperti yang diharapkan.

Ap Payment								
Showing 1-1 of								
Page								
#	Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status
								Status
1	APTRANS0001	APPAYMENT0001	2016-05-31	Expired	52632912500	52632912500	0	Complete

Gambar 5. 19. Data awal *Account Payable Payment*

Create Ap Payment

Ap Transaction ID

Currency ID

Supplier ID

Payment Name

Amount

Date

Paid

Due Date

Residu

Date Completed

Status

Note

Ap Payment Line





#	Date Pay	Payment Line Number	Payment Line Name	Pay	GI Journal
1	Ch			0	Choose GI Journal



57reap

Gambar 5. 20. Proses penginputan data *Account Payable Payment*

Ap Payment								
Showing 1-2 of 2 list								
#	Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status
		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Status
1	APTRANS0001	APPAYMENT0001	2016-05-31	Expired	52632912500	52632912500	0	Complete
2	APTRANS0001	APPAYMENT0002	(not set)	-	52632912500	90000000000	43682912500	Complete

Gambar 5. 21. Data *Account Payable Payment* yang baru berhasil disimpan ke sistem







Payment Name	Amount
Payment Transaksi 1	52632912500
Date	Paid
Choose Date	9000000000
Due Date	Residu
2016-06-30  	43632912500
Date Completed	Status
Choose Date Complete  	INCOMPLETE
Note	
Note	

Ap Payment Line						
#	Date Pay	Payment Line Number	Payment Line Name	Pay	GI journal	
1	 	Payment 1	pembayaran 1	9000000000	Pembelian	
<div>+Add Row</div>						

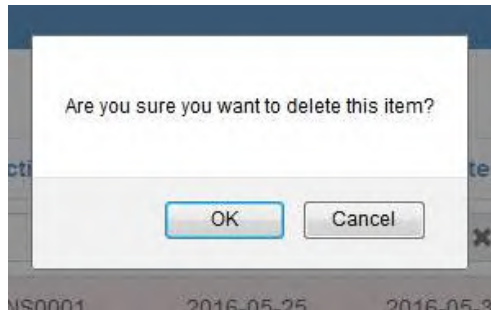
Update

Cancel

Gambar 5. 22. Perubahan data *Account Payable Payment*

Ap Payment								
Showing 1-2 of 2 items								
#	Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status
		<input type="text"/>	 		<input type="text"/>	<input type="text"/>	<input type="text"/>	Status 
1	APTRANS001	APPAYMENT001	2016-05-31	 Expired	52632912500	52632912500	0	 Complete
2	APTRANS001	APPAYMENT002	2016-06-30	18 days	52632912500	9000000000	43632912500	 Incomplete

Gambar 5. 23. Data *Account Payable Payment* berhasil diperbarui



Gambar 5. 24. Proses penghapusan *Account Payable Payment*

Ap Payment								Showing 1-1 of	
#	Transaction Number	Payment Number	Due Date	Time Left	Amount	Paid	Residu	Status	
1	APTRANS0001	APPAYMENT0001	2016-05-31	Expired	52632912500	52632912500	0	Complete	

Gambar 5. 25. *Account Payable Payment* telah terhapus

5.2.4.3. Pengujian Fitur Mengelola Transaksi Penjualan

Pengujian ini terdiri dari pengujian membuat, menyunting, dan menyunting *Account Receivable Transaction*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6.

Tabel 5. 5. Pengujian mengelola transaksi penjualan

ID	UJ-UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian mengelola <i>Account Receivable Transaction</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Account Receivable Transaction</i>
Skenario 1	<i>Pengguna membuat Account Receivable Transaction baru.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> .

Data Uji	Inputan data <i>Account Receivable Transaction</i> :
Langkah Pengujian	Pengguna memilih kegiatan membuat <i>Account Receivable Transaction</i> baru, lalu memasukkan data inputan dan menekan tombol “ <i>Create</i> ”.
Hasil Yang Diharapkan	<i>Account Receivable Transaction</i> yang baru terdapat pada daftar <i>Account Receivable Transaction</i> .
Hasil Yang Didapat	<i>Account Receivable Transaction</i> yang baru terdapat pada daftar <i>Account Receivable Transaction</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> dengan tambahan data <i>Account Receivable Transaction</i> yang baru.
Skenario 2	<i>Pengguna menyunting Account Receivable Transaction tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> .
Data Uji	Perubahan data <i>Account Receivable Transaction</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>Account Receivable Transaction</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>Account Receivable Transaction</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>Account Receivable Transaction</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> .
Skenario 3	<i>Pengguna menghapus Account Receivable Transaction tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> .

Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “Yes” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>Account Receivable Transaction</i> yang baru dihapus tidak tampil pada daftar <i>Account Receivable Transaction</i> .
Hasil Yang Didapat	<i>Account Receivable Transaction</i> yang baru dihapus tidak tampil pada daftar <i>Account Receivable Transaction</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Transaction</i> .

Gambar 5.28 dan Gambar 5.29 menunjukkan proses penginputan data *Account Receivable Transaction* untuk skenario 1. Sedangkan untuk Gambar 5.30 menunjukkan hasil pengujiannya, dimana dapat dilihat penambahan *Account Receivable Transaction* yang baru. Kemudian pada Gambar 5.31 menunjukkan proses merubah data *Account Receivable Transaction*. Hasil pengujiannya dapat dilihat pada Gambar 5.32. Sedangkan pada Gambar 5.33 dapat dilihat proses penghapusan *Account Receivable Transaction*. Gambar 5.34 menunjukkan pada daftar bahwa *Account Receivable Transaction* telah terhapus. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-003 telah bekerja dengan baik seperti yang diharapkan.

The screenshot shows a web application window titled "Ar Transaction" with a status bar indicating "Showing 1-1 of 1". The main area contains a table with the following columns: #, Transaction Number, Date, Due Date, Time Left, and Ar Transaction Status ID. Below the table, there is a list of transactions. The first transaction is highlighted in red and has a status of "Expired".

#	Transaction Number	Date	Due Date	Time Left	Ar Transaction Status ID
1	ARTRANS0001	2016-05-25	2016-05-31		Expired

At the bottom of the window, there are two buttons: "Expired" and "Return".

Gambar 5. 26. Data awal *Account Receivable Transaction*

Create Ar Transaction

Ar Transaction Type ID

Sd Sale ID **Sd Customer ID** **Ar Payterm ID** **Currency ID**

Basic **Detail** **Dunning**

Transaction Name **Amount** **Total**

Date **Due Date** **Note**

Gambar 5. 27. Proses penginputan data *Account Receivable Transaction*

Ar Transaction

Ar Transaction							Showing 1-2 of 2 items.	
#	Transaction Number	Date	Due Date	Time Left	Ar Transaction Status ID	Note		
1	ARTRANS0001	2016-05-25	2016-05-31	Expired	Return			
2	(not set)	2016-06-12	2016-06-30	18 days	Incomplete			

Gambar 5. 28. Data *Account Receivable Transaction* yang baru berhasil disimpan ke sistem

Update Ar Transaction: 2

Ar Transaction Type ID
 New Transaction x ▾

Sd Sale ID Choose Sd sale ▾ **Sd Customer ID** Vicky Chyistian x ▾ **Ar Payterm ID** Standard Payterm x ▾ **Currency ID** IDR

Basic **Detail** **Dunning**

Transaction Name Transaksi New **Amount** 8000000 **Total** 8000000

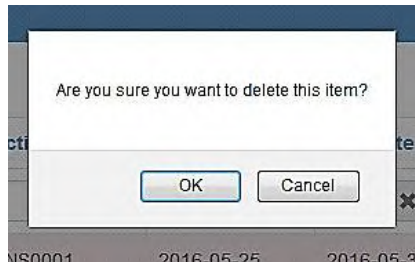
Date 2016-06-12 x 📅 **Due Date** 2016-06-20 x 📅 **Note** Note

Update **Cancel**

Gambar 5. 29. Proses perubahan data *Account Receivable Transaction*

Ar Transaction						Showing 1-2 of 2
						Page ▾ ▮ ▮ ▮ 📄
#	Transaction Number	Date	Due Date	Time Left	Ar Transaction Status ID	
	<input type="text"/>	📅 x	📅 x <input type="text"/>		Status ▾	
1	ARTRANS0001	2016-05-25	2016-05-31	❌ Expired	🔄 Return	
2	(not self)	2016-06-12	2016-06-20	8 days	❌ Incomplete	

Gambar 5. 30. Data *Account Receivable Transaction* berhasil diperbarui



Gambar 5. 31. Proses penghapusan *Account Receivable Transaction*



Gambar 5. 32. *Account Receivable Transaction* telah terhapus

5.2.4.4. Pengujian Fitur Mengelola Pembayaran Penjualan

Pengujian ini terdiri dari pengujian membuat, menyunting, dan menyunting *Account Receivable Payment*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7.

Tabel 5. 6. Pengujian mengelola pembayaran pembelian

ID	UJ-UC-004
Referensi Kasus Penggunaan	UC-004
Nama	Pengujian mengelola <i>Account Receivable Payment</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Account Receivable Payment</i>
Skenario 1	<i>Pengguna membuat Account Receivable Payment baru.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> .
Data Uji	Inputan data <i>Account Receivable Payment</i> :
Langkah Pengujian	Pengguna memilih kegiatan membuat <i>Account Receivable Payment</i> baru, lalu memasukkan data inputan dan menekan

	tombol “Create”.
Hasil Yang Diharapkan	<i>Account Receivable Payment</i> yang baru terdapat pada daftar <i>Account Receivable Payment</i> .
Hasil Yang Didapat	<i>Account Receivable Payment</i> yang baru terdapat pada daftar <i>Account Receivable Payment</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> dengan tambahan data <i>Account Receivable Payment</i> yang baru.
Skenario 2	<i>Pengguna menyunting Account Receivable Payment tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> .
Data Uji	Perubahan data <i>Account Receivable Payment</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>Account Receivable Payment</i> tertentu, lalu melakukan perubahan data dan menekan tombol “Update”.
Hasil Yang Diharapkan	<i>Account Receivable Payment</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>Account Receivable Payment</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> .
Skenario 3	<i>Pengguna menghapus Account Receivable Payment tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> .
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “Yes” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>Account Receivable Payment</i> yang baru dihapus tidak tampil pada daftar <i>Account Receivable Payment</i> .
Hasil Yang Didapat	<i>Account Receivable Payment</i> yang baru dihapus tidak tampil pada daftar <i>Account Receivable Payment</i> .
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>Account Receivable Payment</i> .

Gambar 5.35 menunjukkan proses penginputan data *Account Receivable Payment* untuk skenario 1. Sedangkan untuk Gambar 5.36 menunjukkan hasil pengujiannya, dimana dapat dilihat penambahan *Account Receivable Payment* yang baru. Kemudian pada Gambar 5.37 menunjukkan proses merubah data *Account Receivable Payment*. Hasil pengujiannya dapat dilihat pada Gambar 5.38. Sedangkan pada Gambar 5.39 dapat dilihat proses penghapusan *Account Receivable Payment*. Gambar 5.40 menunjukkan pada daftar bahwa *Account Receivable Payment* telah terhapus. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-004 telah bekerja dengan baik seperti yang diharapkan.

Create Ar Payment

Ar Transaction ID	Ar Payterm ID	Currency ID	Customer ID
Choose Ar transaction	Choose Ar payterm	Choose Currency	Choose Customer

Payment Name	Amount
PaymentName	Amount

Date	Paid
Choose Date	Paid

Due Date	Residu
Choose DueDate	Residu

Date Completed	Status
Choose DateCompleted	Status

Note
Note

Ar Payment Line						
#	Date Pay	Payment Line Number	Payment Line Name	Pay	Gl Journal	Note
1				0	Choose Gl Journal	
<div>+Add Row</div>						

Create

Cancel

Gambar 5. 33. Proses penginputan data *Account Receivable Payment*

15		ARTRANS0014	ARPAYME303105	2016-06-02	2016-07-02	19 days	17322485594	10000000000	7322485594	
16		ARTRANS0016	ARPAYME303106	2016-06-02	2016-06-25	12 days	7071691684	4000000000	3071691684	
17		ARTRANS0017	ARPAYME303107	2016-06-03	2016-06-17	4 days	7759818300	4500000000	3259818300	
18		ARTRANS0020	ARPAYME303108	2016-06-04	2016-06-25	12 days	4853698535	2000000000	2853698535	

Gambar 5. 34. Data Account Receivable Payment yang berhasil disimpan ke sistem

Update Ar Payment: 303109

Ar Transaction ID	Ar Payterm ID	Currency ID	Customer ID
ARTRANS0020	Standard Payterm (fines & discounts) (10	IDR	Nopel2 RE

Payment Name	Amount
Payment Transaksi 20	4853698535

Date	Paid
2016-06-23	1000000000

Due Date	Residu
2016-06-25	3853698535

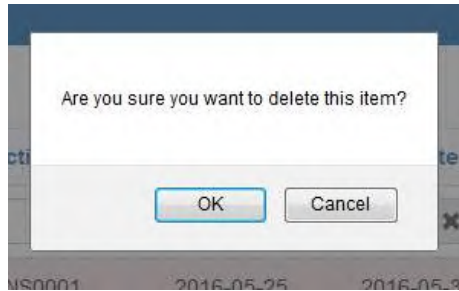
Date Completed	Status
Choose DateCompleted	INCOMPLETE

Note
Nota

Gambar 5. 35. Proses perubahan data Account Receivable Payment

17		ARTRANS0017	ARPAYME303107	2016-06-03	2016-06-17	4 days	7759818300	4500000000	3259818300	
18		ARTRANS0020	ARPAYME303108	2016-06-04	2016-06-25	12 days	4853698535	2000000000	2853698535	
19		ARTRANS0020	ARPAYME303109	2016-06-23	2016-06-30	17 days	4853698535	1000000000	3853698535	

Gambar 5. 36. Data Account Receivable Payment berhasil diperbarui



Gambar 5. 37. Proses penghapusan Account Receivable Payment

16	ARTRANS0016	ARPAYME303106	2016-06-02	2016-06-25	12 days	7071691684	4000000000	3071691684	incomplete
17	ARTRANS0017	ARPAYME303107	2016-06-03	2016-06-17	4 days	7759818300	4500000000	3259818300	incomplete
18	ARTRANS0020	ARPAYME303108	2016-06-04	2016-06-25	12 days	4853698535	2000000000	2853698535	incomplete

Gambar 5. 38. Account Receivable Payment telah terhapus

5.2.4.5. Pengujian Fitur Melakukan Konfirmasi Sales Order

Pengujian ini terdiri pengujian menyetujui dan menolak *Sales Order*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8.

Tabel 5. 7. Pengujian melakukan konfirmasi sales order

ID	UJ-UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian melakukan konfirmasi <i>sales order</i>
Tujuan Pengujian	Menguji kemampuan aplikasi melakukan konfirmasi <i>sales order</i>
Skenario 1	<i>Pengguna menyetujui sales order tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman konfirmasi <i>sales order</i>
Data Uji	Perubahan data pada <i>sales order</i> status:

Langkah Pengujian	Pengguna memilih kegiatan menyetujui <i>sales order</i> dengan menekan tombol “ <i>approve</i> ”
Hasil Yang Diharapkan	<i>Sales Order yang dipilih tidak muncul dalam daftar sales order</i>
Hasil Yang Didapat	<i>Sales Order yang dipilih tidak muncul dalam daftar sales order</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Status <i>Sales Order</i> menjadi disetujui.
Skenario 2	<i>Pengguna menolak Sales Order tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman konfirmasi <i>sales order</i>
Data Uji	Perubahan data pada <i>sales order</i> status:
Langkah Pengujian	Pengguna memilih kegiatan menyetujui <i>sales order</i> dengan menekan tombol “ <i>reject</i> ”.
Hasil Yang Diharapkan	<i>Sales Order yang dipilih tidak muncul dalam daftar sales order</i>
Hasil Yang Didapat	<i>Sales Order yang dipilih tidak muncul dalam daftar sales order</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Status <i>Sales Order</i> menjadi disetujui.

Gambar 5.41 menunjukkan hasil pengujian konfirmasi menyetujui *Sales Order* untuk skenario 1, dimana dapat dilihat *Sales Order* yang telah disetujui tidak muncul dalam daftar *Approve Sales Order*. Sedangkan pada Gambar 5.42 dapat dilihat hasil konfirmasi menolak *Sales Order*, maka *Sales Order* yang dipilih berstatus rejected. Dengan melihat hasil pengujian pada kedua skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-005 telah bekerja dengan baik seperti yang diharapkan.

AR Approve: 12 Jun 2016				
Quotation Number	Customer	Amount	Valid Date	Approve
QO00015	Noal GR	9025939094.7	2016-06-04 00:00:00	

Gambar 5. 39. *Sales Order* berhasil disetujui

AR Approve: 12 Jun 2016					
Quotation Number	Customer	Amount	Valid Date	Approve	Reject
Q060016	Noel CIR	9025939034 7	2016-06-04 00:00:00		<input checked="" type="button" value="Reject"/>

Gambar 5. 40. Sales Order berhasil ditolak

5.2.4.6. Pengujian Fitur Melihat Umur Hutang Perusahaan

Pengujian ini terdiri pengujian melihat umur hutang perusahaan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.9.

Tabel 5. 8. Pengujian melihat umur hutang perusahaan

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian melihat umur hutang perusahaan
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan umur hutang perusahaan.
Skenario 1	<i>Pengguna melihat umur hutang perusahaan.</i>
Kondisi Awal	Pengguna berada pada halaman melihat umur hutang
Data Uji	Tampilan data hutang perusahaan.
Langkah Pengujian	Pengguna memilih lihat umur hutang perusahaan
Hasil Yang Diharapkan	Sistem menampilkan umur hutang perusahaan
Hasil Yang Didapat	Sistem menampilkan umur hutang perusahaan
Hasil Pengujian	Berhasil
Kondisi Akhir	Hutang perusahaan ditampilkan sesuai umurnya.

Gambar 5.43 menunjukkan tampilan umur hutang perusahaan untuk skenario 1. Dengan melihat hasil pengujian pada skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-006 telah bekerja dengan baik seperti yang diharapkan

AP Aging Analysis: 18 Jul 2016									
No	Transaction Number	Amount	Due Date	Current	1-30 Days	31-60 Days	61-90 Days	>90 Days	
Wim Supply									
1	APTRAV50003	IDR 29,000,000.00	2016-06-10	-	-	IDR 29,000,000.00	-	-	
	Sub Total	IDR 29,000,000.00	-			IDR 29,000,000.00			
	Total	IDR 29,000,000.00	-			IDR 29,000,000.00			

Gambar 5. 41. Umur hutang perusahaan berhasil ditampilkan

5.2.4.7. Pengujian Fitur Melihat Umur Piutang Perusahaan

Pengujian ini terdiri pengujian melihat umur piutang perusahaan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.10.

Tabel 5. 9. Pengujian melihat umur piutang perusahaan

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian melihat umur piutang perusahaan
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan umur piutang perusahaan.
Skenario 1	<i>Pengguna melihat umur piutang perusahaan.</i>
Kondisi Awal	Pengguna berada pada halaman melihat umur piutang
Data Uji	Tampilan data piutang perusahaan.
Langkah Pengujian	Pengguna memilih lihat umur piutang perusahaan
Hasil Yang Diharapkan	Sistem menampilkan umur piutang perusahaan
Hasil Yang Didapat	Sistem menampilkan umur piutang perusahaan
Hasil Pengujian	Berhasil
Kondisi Akhir	Piutang perusahaan ditampilkan sesuai umurnya.

Gambar 5.44 menunjukkan tampilan umur piutang perusahaan untuk skenario. Dengan melihat hasil pengujian pada skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-007 telah bekerja dengan baik seperti yang diharapkan.

No	Transaction Number	Amount	Due Date	Current	1-30 Days	31-60 Days	61-90 Days	>90 Days
Wim Supply								
1	APTRAN50003	IDR 29,000,000.00	2015-06-10	-	-	IDR 29,000,000.00	-	-
Sub Total		IDR 29,000,000.00	-			IDR 29,000,000.00		
Total		IDR 29,000,000.00	-			IDR 29,000,000.00		

Gambar 5. 42. Umur Piutang Perusahaan berhasil ditampilkan

5.2.4.8. Pengujian Fitur Melihat Ranking Hutang Perusahaan

Pengujian ini terdiri pengujian melihat ranking hutang perusahaan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.11.

Tabel 5. 10. Pengujian melihat ranking hutang perusahaan

ID	UJ-UC-008
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian melihat ranking hutang perusahaan
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan ranking hutang perusahaan.
Skenario 1	<i>Pengguna melihat ranking hutang perusahaan.</i>
Kondisi Awal	Pengguna berada pada halaman melihat ranking hutang
Data Uji	Tampilan data ranking perusahaan.
Langkah Pengujian	Pengguna memilih lihat ranking hutang perusahaan
Hasil Yang Diharapkan	Sistem menampilkan ranking hutang perusahaan
Hasil Yang Didapat	Sistem menampilkan ranking hutang perusahaan
Hasil Pengujian	Berhasil

Kondisi Akhir	Hutang perusahaan ditampilkan sesuai tingkat kepentingannya.
---------------	--

Gambar 5.45 menunjukkan tampilan ranking hutang perusahaan untuk skenario 1. Dengan melihat hasil pengujian pada skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-008 telah bekerja dengan baik seperti yang diharapkan.

Debt Analysis									
Transaction Number	Transaction Name	Due Date	Fine Per Month(IDR)	Total Debt(IDR)	Payment Tolerance(Days)	Time Left (Days)	Late Frequency(%)	Discount Early Payment(IDR)	Scores
APTRANS0003	Transaction PO 13	2016-06-10	1450000	IDR 29,000,000.00	7	8	0	IDR 600,000.00	0.148
APTRANS0002	Transaction PRO 14	2016-08-12	0	IDR 0.00	7	25	0	IDR 4,613,750.00	0.296
APTRANS0001	Transaksi 1	2016-07-31	0	IDR 0.00	7	13	0	IDR 49,000,000.00	0.373

Gambar 5. 43. Ranking hutang perusahaan berhasil ditampilkan

5.2.4.9. Pengujian Fitur Melihat Ranking Pelanggan

Pengujian ini terdiri pengujian ranking pelanggan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.12.

Tabel 5. 11. Pengujian melihat ranking pelanggan

ID	UJ-UC-009
Referensi Kasus Penggunaan	UC-009
Nama	Pengujian melihat ranking pelanggan
Tujuan Pengujian	Menguji kemampuan aplikasi menampilkan ranking pelanggan.
Skenario 1	Pengguna melihat ranking pelanggan.
Kondisi Awal	Pengguna berada pada halaman melihat ranking pelanggan
Data Uji	Tampilan data ranking pelanggan.
Langkah	Pengguna memilih lihat ranking pelanggan

Pengujian	
Hasil Yang Diharapkan	Sistem menampilkan ranking pelanggan
Hasil Yang Didapat	Sistem menampilkan ranking pelanggan
Hasil Pengujian	Berhasil
Kondisi Akhir	Ranking pelanggan ditampilkan sesuai <i>score</i> pembelian yang pernah dilakukan.

Gambar 5.46 menunjukkan tampilan ranking pelanggan untuk skenario 1. Dengan melihat hasil pengujian pada skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC-009 telah bekerja dengan baik seperti yang diharapkan.



The screenshot shows a web application interface titled "Customer Analysis". It includes a "Choose Customer Group" dropdown menu with "Enterprise" selected and a "Submit" button. Below this, the text "Customer Group: Enterprise" is displayed. A table shows the ranking of customers based on various financial metrics. The table has 8 columns: Customer Name, Late Frequency (%), Bad Debts (IDR), Paid Before Maturity (%), Total Payment (IDR), Payment Complete Average(Month), Order Frequency, and Score. One customer, I Gede Arya Putra Perdana, is listed with a score of 1.

Customer Name	Late Frequency (%)	Bad Debts (IDR)	Paid Before Maturity (%)	Total Payment (IDR)	Payment Complete Average(Month)	Order Frequency	Score
I Gede Arya Putra Perdana	0	0.00	0	1.200.848.000.00		0	1

Gambar 5. 44. Ranking pelanggan berhasil ditampilkan

5.3. Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi pengujian ketertarikan partisipan terhadap aplikasi.

5.3.1. Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.13. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan

baik. Sehingga dapat ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5. 12. Rangkuman Hasil Pengujian

ID	Nama	Skenario	Hasil
UJ-UC-001	Pengujian mengelola transaksi pembelian	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-002	Pengujian mengelola pembayaran pembelian	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-003	Pengujian mengelola transaksi penjualan	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-004	Pengujian mengelola pembayaran penjualan	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-005	Pengujian melakukan konfirmasi <i>Sales Order</i>	Skenario 1	Berhasil
		Skenario 2	
UJ-UC-006	Pengujian melihat umur hutang perusahaan	Skenario 1	Berhasil
UJ-UC-007	Pengujian melihat umur piutang perusahaan	Skenario 1	Berhasil
UJ-UC-008	Pengujian melihat ranking hutang perusahaan	Skenario 1	Berhasil
UJ-UC-009	Pengujian melihat ranking pelanggan	Skenario 1	Berhasil

5.3.2. Evaluasi Pengujian Perhitungan Metode

Pengujian dilakukan dengan menggunakan data transaksi perusahaan pada simulasi ERP kelas sistem enterprise. Dari 5 perusahaan yang ada, diambil 3 perusahaan dengan data transaksi terbaik. Pada pengujian sistem perankingan hutang, yang digunakan adalah data transaksi perusahaan A, B, dan C. Hasil pengujian perhitungan metode perankingan hutang perusahaan A, B, dan C secara manual dan hasil perhitungan pada aplikasi ditunjukkan pada tabel 5.14, 5.15, dan 5.16. Berdasarkan data pada tabel tersebut, antara perhitungan manual dan perhitungan pada aplikasi menunjukkan hasil perhitungan dengan urutan yang sama.

Tabel 5. 13. Perbandingan hasil perhitungan ranking hutang perusahaan A

Perhitungan Manual		Perhitungan Aplikasi	
Transaction Number	Score	Transaction Number	Score
APTRANS0029	0.027	APTRANS0029	0.027
APTRANS0012	0.031	APTRANS0012	0.030
APTRANS0031	0.036	APTRANS0031	0.036
APTRANS0011	0.037	APTRANS0011	0.037
APTRANS0037	0.044	APTRANS0037	0.044
APTRANS0033	0.045	APTRANS0033	0.045
APTRANS0034	0.046	APTRANS0034	0.046
APTRANS0032	0.047	APTRANS0032	0.047
APTRANS0036	0.048	APTRANS0036	0.048
APTRANS0035	0.068	APTRANS0035	0.068
APTRANS0025	0.069	APTRANS0025	0.069
APTRANS0009	0.071	APTRANS0009	0.070

Perhitungan Manual		Perhitungan Aplikasi	
APTRANS0028	0.080	APTRANS0028	0.079
APTRANS0018	0.091	APTRANS0018	0.091
APTRANS0038	0.105	APTRANS0038	0.105
APTRANS0034	0.155	APTRANS0034	0.155

Tabel 5. 14. Perbandingan hasil perhitungan ranking hutang perusahaan B

Perhitungan Manual		Perhitungan Aplikasi	
Transaction Number	Score	Transaction Number	Score
APTRANS0004	0.076	APTRANS0004	0.076
APTRANS0013	0.095	APTRANS0013	0.09
APTRANS0011	0.098	APTRANS0011	0.093
APTRANS0015	0.103	APTRANS0015	0.098
APTRANS0014	0.108	APTRANS0014	0.103
APTRANS0012	0.108	APTRANS0012	0.104
APTRANS0010	0.114	APTRANS0010	0.114
APTRANS0003	0.136	APTRANS0003	0.136
APTRANS0005	0.16	APTRANS0005	0.155

Tabel 5. 15. Perbandingan hasil perhitungan ranking hutang perusahaan C

Perhitungan Manual		Perhitungan Aplikasi	
Transaction Number	Score	Transaction Number	Score
APTRANS0024	0.015	APTRANS0024	0.015
APTRANS0029	0.015	APTRANS0029	0.015

Perhitungan Manual		Perhitungan Aplikasi	
APTRANS0015	0.016	APTRANS0015	0.016
APTRANS0002	0.016	APTRANS0002	0.016
APTRANS0028	0.017	APTRANS0028	0.017
APTRANS0011	0.017	APTRANS0011	0.017
APTRANS0027	0.018	APTRANS0027	0.018
APTRANS0016	0.019	APTRANS0016	0.020
APTRANS0022	0.021	APTRANS0022	0.022
APTRANS0017	0.026	APTRANS0017	0.024
APTRANS0004	0.026	APTRANS0004	0.025
APTRANS0021	0.026	APTRANS0021	0.026
APTRANS0026	0.026	APTRANS0026	0.026
APTRANS0033	0.027	APTRANS0033	0.027
APTRANS0025	0.028	APTRANS0025	0.028
APTRANS0013	0.029	APTRANS0013	0.029
APTRANS0019	0.030	APTRANS0019	0.030
APTRANS0020	0.031	APTRANS0020	0.031
APTRANS0035	0.033	APTRANS0035	0.033
APTRANS0018	0.034	APTRANS0018	0.034
APTRANS0008	0.036	APTRANS0008	0.036
APTRANS0007	0.037	APTRANS0007	0.037
APTRANS0003	0.044	APTRANS0003	0.044
APTRANS0023	0.056	APTRANS0023	0.056
APTRANS0040	0.062	APTRANS0040	0.062
APTRANS0034	0.079	APTRANS0034	0.079

Perhitungan Manual		Perhitungan Aplikasi	
APTRANS0032	0.205	APTRANS0032	0.205

Pada pengujian sistem perankingan pelanggan, yang digunakan adalah data pelanggan wholesale perusahaan A, B dan C. Hasil pengujian perhitungan metode perankingan pelanggan wholesale perusahaan A, B dan C secara manual dan hasil perhitungan pada aplikasi ditunjukkan pada tabel 5.17, 5.18, 5.19.

Tabel 5. 16. Perbandingan hasil perhitungan ranking pelanggan Perusahaan A

Perhitungan Manual		Perhitungan Aplikasi	
Customer Number	Score	Customer Number	Score
Nopal RE	0.432	Nopal RE	0.442
Noal GR	0.292	Noal GR	0.292
Nopil ME	0.275	Nopil ME	0.275

Tabel 5. 17. Perbandingan hasil perhitungan ranking pelanggan Perusahaan B

Perhitungan Manual		Perhitungan Aplikasi	
Customer Number	Score	Customer Number	Score
Franky Setiawan	0.406	Franky Setiawan	0.406
Budi Pangestu Tanujaya	0.309	Budi Pangestu Tanujaya	0.308
Gian Sebastian	0.178	Gian Sebastian	0.178
Ayu Kardina Sukmawati	0.107	Ayu Kardina Sukmawati	0.107

Tabel 5. 18. Perbandingan hasil perhitungan ranking pelanggan Perusahaan C

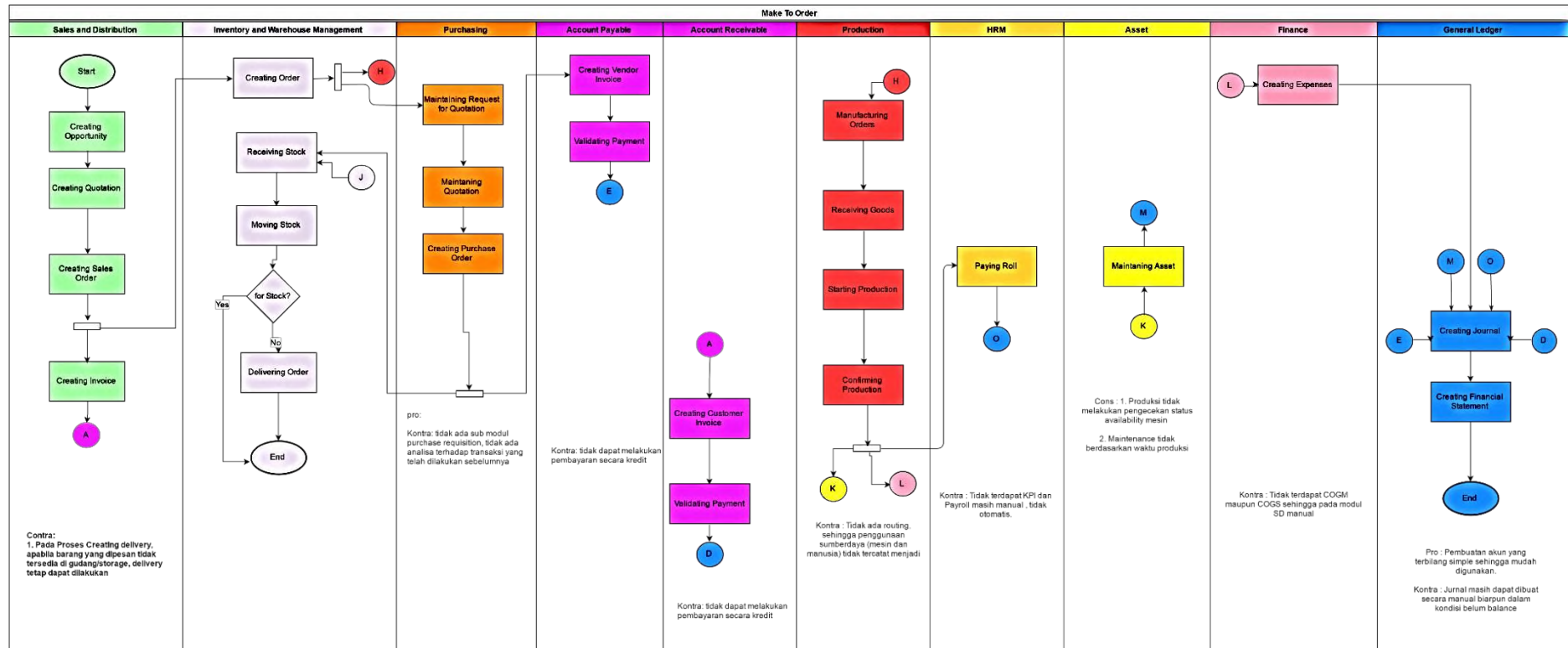
Perhitungan Manual		Perhitungan Aplikasi	
Customer Number	Score	Customer Number	Score
Kelompok 0 Medium Wholesaler	0.264	APTRANS0004	0.263
Kelompok 0 Grosir Wholesaler	0.252	APTRANS0013	0.252
Kelompok 0 Retail Wholesaler	0.142	APTRANS0011	0.142
A	0.115	APTRANS0015	0.117
C	0.113	APTRANS0014	0.114
B	0.109	APTRANS0012	0.109

Berdasarkan data pada tabel-tabel diatas, antara perhitungan manual dan perhitungan pada aplikasi menunjukkan hasil perhitungan dan urutan yang sama. Sehingga dapat ditarik kesimpulan bahwa perhitungan metode dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

[Halaman ini sengaja dikosongkan]

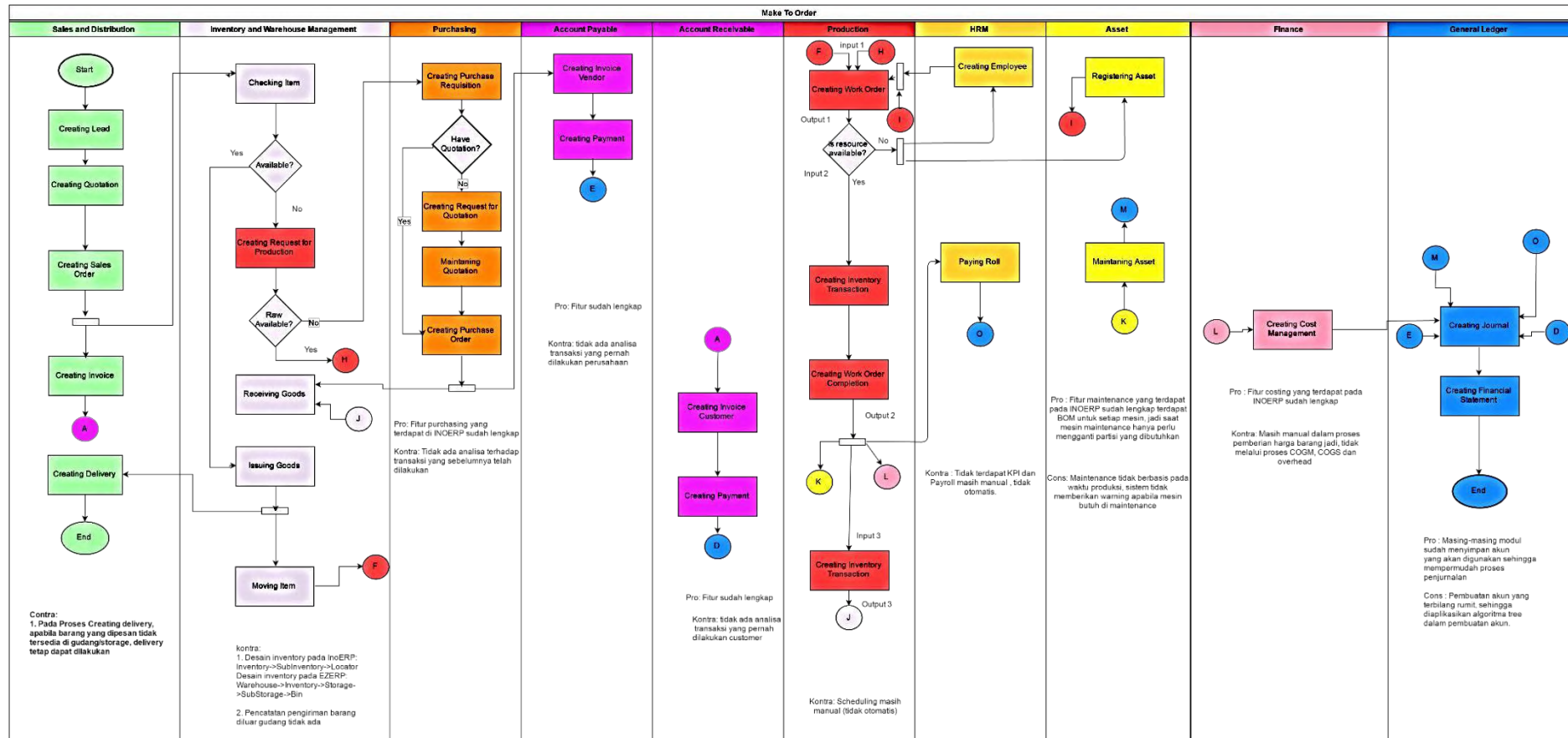
LAMPIRAN A – PROSES BISNIS

A.1. Proses Bisnis Odoo



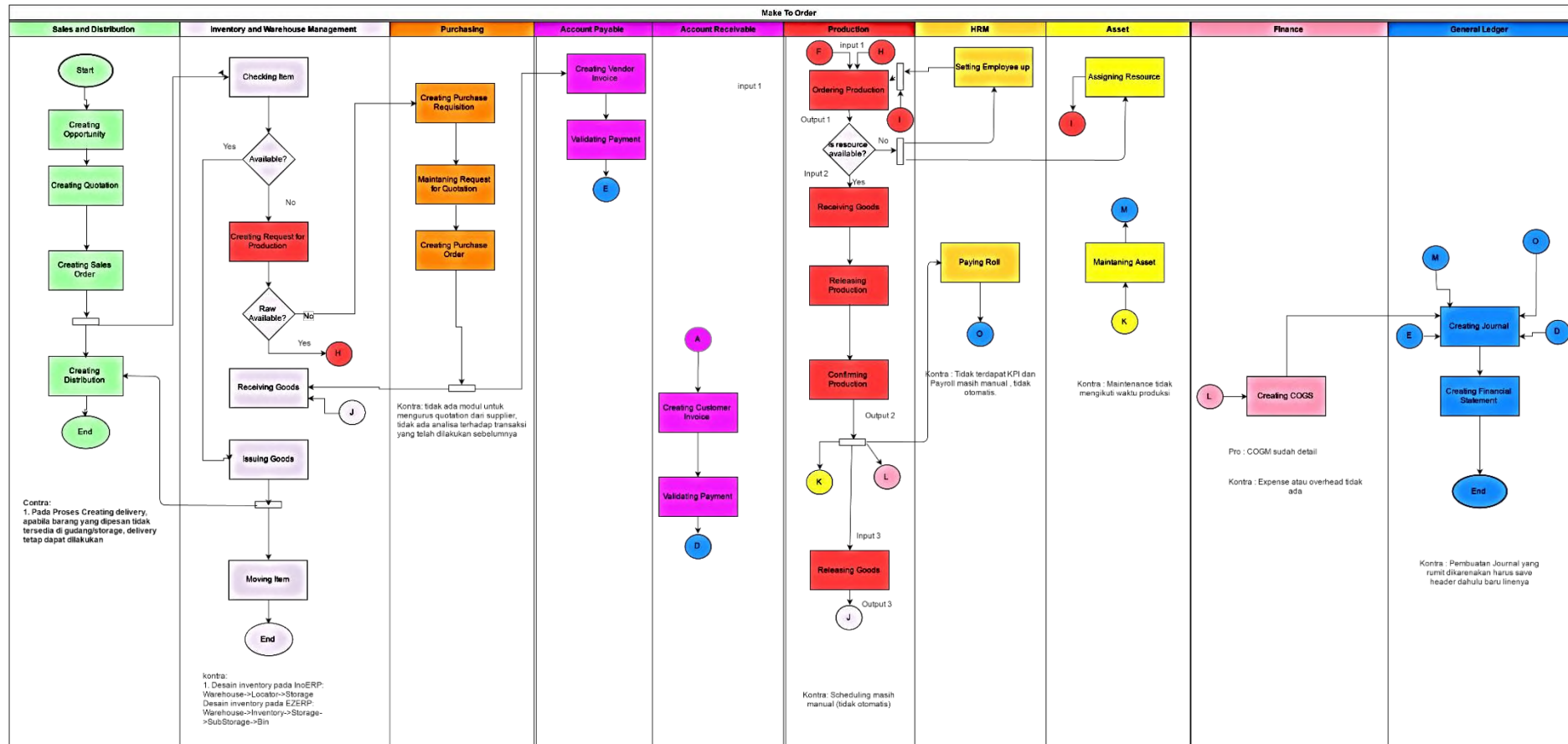
Lampiran A. 1. Proses Bisnis Odoo

A.2 Proses Bisnis InoERP



Lampiran A. 2. Proses Bisnis InoERP

A.3 Proses Bisnis Adempiere



Lampiran A. 3. Proses Bisnis Adempiere

LAMPIRAN B.1 – FORM KUESIONER

FORM KUESIONER PENENTUAN KRITERIA

KUESIONER PENELITIAN Judul Penelitian RANCANG BANGUN SISTEM <i>ENTERPRISE RESOURCE PLANNING</i> PADA MODUL <i>ACCOUNT PAYABLE</i> DAN <i>ACCOUNT RECEIVABLE</i> YANG BERORIENTASI <i>MULTI-TENANCY</i> DENGAN <i>DISTRIBUTED DATABASE</i>							
<p>Bapak/Ibu/Sdr/i yang saya hormati,</p> <p>Saya Syafiyus Kurniawati, Mahasiswi S-1 Teknik Informatika, Institut Teknologi Sepuluh Nopember. Dalam hal ini saya sedang mengadakan penelitian Tugas Akhir. Kuesioner ini berhubungan dengan tanggapan Anda terhadap perankingan hutang dan pelanggan pada suatu perusahaan. Hasil kuesioner ini tidak untuk dipublikasikan, melainkan untuk kepentingan penelitian semata.</p> <p>Atas bantuan, kesediaan waktu, dan kerjasama Bapak/Ibu/Sdr/i, sebelum dan sesudahnya saya ucapkan terima kasih.</p> <p>Profil Responden</p> <p>Nama : _____</p> <p>Pekerjaan : _____</p> <p>❖ Petunjuk Pengisian Kuesioner</p> <p>Pada kuesioner ini, di simulasikan terdapat perusahaan yang ingin melakukan perankingan hutang dan perankingan pelanggan yang telah melakukan transaksi pembelian pada perusahaan tersebut.</p> <p>Berilah urutan prioritas (1/2/3/4/5/6) pada kolom jawaban kriteria yang Anda pilih.</p> <p>Serta berilah tanda silang (x) pada kolom jawaban kepentingan dari kriteria yang Anda pilih.</p> <p>Contoh pengisian kuesioner :</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 60%;">Kriteria</th> <th style="width: 30%;">Kepentingan</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; border: 1px solid black; width: 30px;">2</td> <td>Frekuensi customer terlambat membayar.</td> <td style="text-align: center;">(1) x (3) (4)</td> </tr> </tbody> </table> <p>❖ Pertanyaan Kuesioner</p> <ul style="list-style-type: none"> • Apa saja yang menjadi kriteria Anda dalam menentukan perankingan hutang dan 			Kriteria	Kepentingan	2	Frekuensi customer terlambat membayar.	(1) x (3) (4)
	Kriteria	Kepentingan					
2	Frekuensi customer terlambat membayar.	(1) x (3) (4)					

perankingan pelanggan pada suatu perusahaan.

Keterangan :

- (1) Tidak Berpengaruh
- (2) Sedikit Berpengaruh
- (3) Berpengaruh
- (4) Sangat Berpengaruh

Kriteria perankingan hutang yang dianggap paling penting dan paling mendesak.

Kriteria	Kepentingan
<input type="checkbox"/> Toleransi waktu keterlambatan pembayaran hutang yang diberikan oleh supplier.	(1) (2) (3) (4)
<input type="checkbox"/> Sisa waktu pembayaran hutang mendekati jatuh tempo.	(1) (2) (3) (4)
<input type="checkbox"/> Besar diskon apabila melakukan pelunasan lebih awal.	(1) (2) (3) (4)
<input type="checkbox"/> Denda setiap bulan apabila terlambat membayar hutang tersebut.	(1) (2) (3) (4)
<input type="checkbox"/> Frekuensi keterlambatan pembayaran yang pernah dilakukan pada supplier tertentu	(1) (2) (3) (4)
<input type="checkbox"/> Jumlah/besar hutang	(1) (2) (3) (4)

Kriteria perankingan customer menurut tingkat loyalitas customer.

Kriteria	Kepentingan
<input type="checkbox"/> Frekuensi keterlambatan pembayaran yang pernah dilakukan oleh customer.	(1) (2) (3) (4)
<input type="checkbox"/> Jumlah hutang customer yang belum dibayar..	(1) (2) (3) (4)
<input type="checkbox"/> Presentase customer melakukan pelunasan tagihan sebelum tanggal jatuh tempo.	(1) (2) (3) (4)
<input type="checkbox"/> Total Order yang telah dibayar oleh customer	(1) (2) (3) (4)
<input type="checkbox"/> Rata-rata waktu pelunasan tagihan customer.	(1) (2) (3) (4)
<input type="checkbox"/> Frekuensi order customer.	(1) (2) (3) (4)

LAMPIRAN B.2. PROFIL RESPONDEN KUESIONER

No	Nama	Pekerjaan
1.	Agnes Ayuningtyas	Pegawai Akuntansi Jiwasraya
2.	Evi Vidiyanti	Mahasiswi Akuntansi Perbanas
3.	Artantya Eka	Mahasiswi Akuntansi Unair
4.	Risky Pradanta	Swasta
5.	Feri Tri Andi	Mahasiswa STAN
6.	Veny Lidiawati	Karyawan BUMN
7.	Eva Nur Fadilah	Wirausaha
8.	Shinta	Pegawai Bank
9.	Rian Tri Widiyanto	Karyawan BUMN
10	Mia Yuli Setiani	Mahasiswi

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Pengembangan sistem ERP pada modul *Account Payable* dan *Account Receivable* berhasil dimodelkan dengan proses bisnis yang meliputi sub-modul *AP Transaction*, *AP Payment*, *AP Report*, *AP Debt Analysis*, *AR Transaction*, *AR Payment*, *AR Customer Analysis*.
2. *Multitenancy* memungkinkan pengguna untuk bisa mengakses aplikasi yang sama akan tetapi dengan data dari masing-masing *tenant*, Dalam pengaplikasian *multitenancy* berhasil dibantu dengan adanya *Role Based Access Control (RBAC)* untuk dapat membedakan kepemilikan data. Basis data terdistribusi digunakan sebagai solusi untuk dapat menyelesaikan masalah ketika salah satu *tenant* mengalami *fail-over* dan *tenant* lain tetap dapat mengakses aplikasi, dengan menggunakan replikasi dan fragmentasi yang disediakan oleh *MySQL Cluster*.
3. Sistem Perankingan hutang dan pelanggan menggunakan metode AHP (*Analytical Hierarchy Process*) berhasil memudahkan perusahaan dalam menentukan prioritas kepentingan hutang dan prioritas pelanggan terbaik.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Menggunakan metode pengambilan keputusan selain *Analytical Hierarchy Process* (AHP) untuk membandingkan tingkat keakuratan dari setiap metode pengambilan keputusan yang ada.
2. Menggunakan data transaksi perusahaan secara nyata sebagai bahan untuk melakukan analisis.
3. Meningkatkan *running process* untuk proses dengan data yang sangat besar.

DAFTAR PUSTAKA

- [1] "Wikipedia," Wikipedia, 16 Maret 2016. [Online]. Available: https://id.wikipedia.org/wiki/Perencanaan_sumber_daya_perusahaan. [Accessed 9 Juni 2016].
- [2] Y. Adrian, "Insan Performa," 29 February 2016. [Online]. Available: http://www.insanperforma.co.id/index.php?option=com_content&view=article&id=145%3Asistem-erp-dalam-perusahaan&catid=38%3Anews&Itemid=18&lang=. [Accessed 2 Juni 2016].
- [3] P. and . I. G. Budi Rinanta, "IMPLEMENTASI MySQL CLUSTER PADA BASIS DATA TERDISTRIBUSI," *IMPLEMENTASI MySQL CLUSTER PADA BASIS DATA TERDISTRIBUSI*, p. 2, 2012.
- [4] R. W. Herlambang, R. Sarno and D. Sunaryono, "Implementasi Modul-Modul Enterprise Resource Planning Multi-Tenant pada Cloud Computing," *JURNAL TEKNIK POMITS*, vol. 2, 2013.
- [5] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sciences*, vol. 1, 2008.
- [6] S. and M. Ramdhani, "Sistem Pendukung Keputusan," 1998.
- [7] Q. Li, X. Zang and M. Zhu, "Towards secure dynamic collaborations with group-based RBAC model," *Computers & Security*, vol. 26, p. Pages 260–275, 2009.
- [8] R. S. Sandu, "Role-based Access Control," *Advances in Computers*, p. Pages 237–286, 1998.
- [9] T. Reuters, "Accounts Payable Optimization in Today's Healthcare Environment," in *Accounts Payable Optimization in Today's Healthcare Environment*, Canada, Global Healthcare Exchange, 2010.
- [10] L. Torki, R. D. Isfahani and A. Rezai, "Ranking of Financial and Electronic Debts Using Analytic Hierarchy Process," *New Marketing Research Journal*, no. Special, pp. 21-30, 2013.
- [11] "Student Portal," PalcomTech, [Online]. Available: [http://news.palcomtech.com/contoh-terapan-perhitungan-manual-dan-](http://news.palcomtech.com/contoh-terapan-perhitungan-manual-dan)

- analisa-hasil-kuesioner-menggunakan-skala-likert/. [Accessed 27 7 2016].
- [12] F. A. Haery, H. Ghorbani and A. A. Farahmand, "An AHP Approach for Ranking Critical Success Factors of Customers," *International Journal of Marketing Studies*, vol. 6, no. ISSN 1918-719X E-ISSN 1918-7203, 2014.
 - [13] L. Ardiyaningrat and I. G. A. Purnamawati, "ANALISIS TINGKAT PERPUTARAN PIUTANG DAGANG PADA PT. TIRTA MUMBUL JAYA ABADI PERIODE 2010 – 2012," *VOKASI Jurnal Riset Akuntansi*, vol. 2 No 2, no. ISSN 2337 – 537x, Oktober 2013.
 - [14] S. Chen, T. Jian and H. Yang, "A Fuzzy AHP Approach for Evaluating Customer," *JOURNAL OF COMPUTERS*, vol. 6 No 2, 2011.
 - [15] "The Accounts Payable Aging Summary Report," RPPC, 2013. [Online]. Available: <http://rppc.net/quickbooks-quick-tip-the-accounts-payable-aging-summary-report/>. [Accessed 1 July 2016].
 - [16] "Penyakit Vertigo," Penyakit Vertigo, [Online]. Available: <http://penyakitvertigo.com/>.
 - [17] M. Timothy C. Hain, "Dizziness and Balance Exercise," [Online]. Available: <http://www.dizziness-and-balance.com/treatment/rehab/cawthorne.html>. [Accessed 17 05 2015].
 - [18] M. James, "I Programmer," [Online]. Available: <http://www.i-programmer.info/ebooks/practical-windows-kinect-in-c/3725-getting-started-with-windows-kinect-sdk-10.html?start=1>. [Accessed 14 03 2015].
 - [19] E. Cowles, "Brain and Spine," [Online]. Available: <http://www.brainandspine.org.uk/vestibular-rehabilitation-exercises>. [Accessed 21 05 2015].
 - [20] Microsoft, Microsoft Corp., [Online]. Available: <https://www.microsoft.com/en-us/kinectforwindows/>. [Accessed 20 05 2015].
 - [21] Microsoft, Microsoft Corp, [Online]. Available: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>. [Accessed

- 15 05 2015].
- [22] Microsoft, Microsoft Corp, [Online]. Available: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. [Accessed 15 05 2015].
 - [23] A. Rizzo, M.-C. Huang, M.-C. Su, P.-C. Wang, P.-Y. Tsai, S.-C. Yeh and T.-Y. Fang, "Machine learning-based assessment tool for imbalance and vestibular dysfunction with virtual reality rehabilitation system," *elsevier*, p. 312, 2014.
 - [24] ALODOKTER, "ALODOKTER," [Online]. Available: <http://www.alodokter.com/vertigo/pengobatan>. [Accessed 07 05 2015].
 - [25] Monika and Rika, "Calon Apoteker Bercerita," 02 06 2011. [Online]. Available: <https://apotekerbercerita.wordpress.com/2011/06/02/efek-samping-obat/>. [Accessed 10 06 2015].
 - [26] G. Williams, Implementing SAP ERP Sales & Distribution, McGraw-Hill Education, 2008.
 - [27] S. R. Magal and J. Word, Integrated Business Processes with ERP Systems, New Jersey: Wiley and Sons, Inc., 2012.
 - [28] R. J. Hyndman and G. Athanasopoulos, Forecasting: principles and practice, otext.com, 2013.
 - [29] H. Grubb and A. Mason, "Long lead-time forecasting of UK air passengers by Holt–Winters methods with damped trend," *International Journal of Forecasting*, vol. 17, no. 1, pp. 71-82, 2001.
 - [30] "Icmrindia," 2 6 2016. [Online]. Available: <http://www.icmrindia.org/courseware/Sales-and-Distribution-Management1/SDM-Contents.htm>.
 - [31] T. K. Panda and S. Sahadev, Sales and Distribution Management, Oxford: Oxford University Press, 2005.
 - [32] "Pi Accounting," 12 6 2016. [Online]. Available: <http://sistem-akuntansi.com/sistem-akuntansi/sales-order/>.
 - [33] A. V. Allorerung, Rancang Bangun Aplikasi Berorientasi Arsitektur Service (SOA) dengan Pendekatan Workflow pada Domain Customer

- Relationship Management (CRM) untuk Sistem Enterprise Resource Planning, Surabaya: Institut Teknologi Sepuluh Nopember, 2013.
- [34] M. T. Özsu and P. Valduriez, Principles of Distributed Database Systems, New York: Springer, 2011.
- [35] K. Roebuck, Multitenancy: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors, Queensland: Emereo Publishing, 2012.
- [36] D. Ferraiolo, D. R. Kuhn and R. Chandramouli, Role-based Access Control, Artech House, 2003.

DAFTAR ISTILAH

invoice

dokumen yang digunakan sebagai pernyataan tagihan yang harus dibayar oleh customer.

sales order

dokumen yang digunakan untuk mencatat pesanan dari pelanggan.

customer

Pembeli yang melakukan pembelian suatu produk perusahaan, yang biasa disebut pelanggan.

supplier

Perusahaan yang memasok kebutuhan material perusahaan lain.

akun

sistem yang digunakan akuntan untuk mencatat aliran uang dalam perusahaan.

ERP

aplikasi yang digunakan untuk merencanakan dan mengelola sumber daya perusahaan meliputi dana, manusia, mesin, material dan kapasitas produksi yang berpengaruh luas mulai dari manajemen paling atas hingga operasional di sebuah perusahaan agar dapat dimanfaatkan secara optimal.

Odoo

Salah satu aplikasi ERP yang bersifat *open source*.

InoERP

Salah satu aplikasi ERP yang bersifat *open source*.

simulasi bisnis

permainan bisnis yang dirancang agar peserta mengalami dan berlatih menjalankan suatu perusahaan.

SQL

bahasa yang dipergunakan untuk mengakses data dalam basis data

[Halaman ini sengaja dikosongkan]

INDEKS

- account receivable, 66, 142, 143
- AHP, viii, x, 2, 3, 4, 11, 12, 20, 21, 23, 40, 53, 65, 66, 210
- akun**, 223
- BI**, 172
- customer, 1, 3, 4, 13, 30, 51, 137, 143, 144, 161, 223
- debt, 39, 126, 136
- ERP, 29, 30, 63, 209, 223
- hutang, 1, 2, 3, 4, 11, 13, 36, 39, 40, 47, 48, 49, 50, 51, 52, 59, 63, 65, 66, 69, 71, 85, 86, 88, 89, 105, 106, 125, 126, 132, 134, 135, 142, 143, 152, 153, 161, 162, 197, 198, 199, 200, 202, 203, 204
- invoice**, 223
- Multitenancy*, 209
- Odoo**, 223
- payment, 39, 47, 49, 51, 125, 126, 130, 131, 132, 133, 134, 135, 142, 143
- perusahaan**, 223
- piutang, 1, 3, 11, 14, 51, 52, 59, 63, 67, 69, 71, 86, 87, 113, 198, 199, 202
- purchase, 66, 223
- sales order**, 223
- simulasi bisnis**, 223
- SQL**, 223
- transaction, 125, 126, 127, 128, 130, 131, 132, 134, 135, 136, 138, 139, 141, 142, 143

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, **Syafiyus Kurniawati**, lahir di Surabaya, 19 Januari 1994. Penulis menempuh pendidikan sekolah dasar di SD Negeri Asemrowo 1 Surabaya. Melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 3 Surabaya dan penulis melanjutkan sekolah menengah atas di SMA Negeri 6 Surabaya. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi,

Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif dalam organisasi tingkat jurusan. Penulis sempat diberi amanah untuk menjadi asisten dosen Mata Kuliah Sistem Teknologi dan Informasi 2012-2013

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi(MI) dan memiliki ketertarikan di bidang *Data Analyst*. Penulis dapat dihubungi melalui email: syafiyus.k@gmail.com.